AN APPLICATION OF FAULT TREE ANALYSIS TO OPERATIONAL TESTING

A THESIS

Presented to

The Faculty of the Division

of Graduate Studies

By

Gordon Lee Rankin

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Operations Research

Georgia Institute of Technology

June 1975

AN APPLICATION OF FAULT TREE ANALYSIS

TO OPERATIONAL TESTING

Approved:

Harrison M. Wadsworth, Chairman

Leslie G. Callahan Jr.

Russell G. Heikes

Date approved by Chairman: 6/9/75

ACKNOWLEDGMENTS

To recognize everyone whose efforts contributed to making this thesis possible would be an overwhelming task. Special recognition is due the following:

Dr. Harrison M. Wadsworth, chairman of my thesis committee, who patiently listened, gently prodded, and provided invaluable assistance.

Dr. Leslie G. Callahan and Dr. Russell G. Heikes, members of my reading committee, who provided their valuable professional support.

The United States Army, which afforded me the opportunity to attend graduate school at this distinguished university.

The Operational Test and Evaluation Agency of the United States Army, who sponsored the entire effort.

My wife, Jane, who has endured and supported me throughout.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS ii

LIST OF ILLUSTRATIONS iv

SUMMARY. v

Chapter

I. INTRODUCTION 1

Definition of the Problem
Brief History Leading to the Problem
Purpose of the Research
Review of the Literature
Organization of the Research

II. FAULT TREE ANALYSIS 12

General
Fault Tree Terminology and Symbology
System Definition
Fault Tree Construction
Qualitative Analysis

III. A DILEMMA 46

Resolution for Component Importance
The Problem with Cut Set Importance
A Proposal

IV. AN EXAMPLE 58

V. CONCLUSIONS, LIMITATIONS, AND RECOMMENDATIONS 73

APPENDIX

A. QUANTITATIVE ANALYSIS 76

BIBLIOGRAPHY 79

LIST OF ILLUSTRATIONS

Figure		Page
1.	Fault Tree Logic Symbols	16
2.	Fault Tree Event Symbols	17
3.	SIMGUN Schematic.	20
4.	SIMGUN Fault Tree	25
5.	AND and OR Gates.	26
6.	Coded SIMGUN Fault Tree	27
7.	Dual SIMGUN Fault Tree.	37
8.	Dual Fault Tree	49
9.	Command and Control Schematic	59
10.	Fault Tree for C & C System	61
11.	Continuation of Fault Tree for C & C System.	62
12.	MIOD Failure Fault Tree	64
13.	A Separate Power Generating System.	71

SUMMARY

The problem of designing an operational test for complex military systems is approached using fault tree analysis. Operational testing, as opposed to developmental testing, must encompass all the various systems, doctrines, organizations, hardwares, and personnel that impact upon the system. Fault tree analysis is suggested as a method of modeling the entire system for various critical issues.

Algorithms for determining minimal cut sets and minimal path sets are demonstrated. The structural importances of the various components and cut sets are determined analytically using the probabilistic expansion of the union of the minimal path sets as the reliability function. For manual solutions, a simpler sensitivity analysis is demonstrated for component importance, and the author proposes a similar extension to minimal cut sets.

The identification and ranking of components and subsystems permits the testing agency to direct their tests towards the weak links in the system. System weaknesses that might otherwise have been overlooked in testing only to surface after full scale adoption by the U.S. Army can be observed. The causes of important failures can be explored in testing. Fault tree analysis increases system understanding. It provides a vehicle to explore system

alternatives. Structural analysis frees the method from reliability data which may not be available due to the nature of the system or the equipment state of the art.

Fault tree analysis is admittedly a binary modeling scheme which ignores partial failures. It is also situation specific; that is each tree is constructed only about the failure of interest. Hence, it can explore only one critical issue for each model developed.

Despite these weaknesses, fault tree analysis appears to be an applicable modeling technique to determine the weak links, both human and hardware, in a complex system, thereby improving system understanding and test design.

CHAPTER I

INTRODUCTION

The independent evaluations of major U.S. Army developmental systems are conducted by the Operational Test and Evaluation Agency to insure objective, unbiased evaluations of new systems prior to adoption, and full-scale employment by the Army. These evaluations consist of both operational and developmental testing. Developmental testing is primarily the comparison of actual equipment performance with contractual specifications or technical requirements. Functional or developmental testing as a consequence is fairly straightforward.

Operational testing and evaluation, on the other hand, is conducted to determine, insofar as possible, whether the complete system is capable of fulfilling operational requirements. Operational testing must give consideration to interfaces with other systems, tactics, organizational doctrine, and the skills and frailties of its ultimate human users. Operational testing must consider the most probable environment and conditions in which the system will be used which may not necessarily coincide with the requirements for which the system was developed. After all a perfectly functioning system which cannot cope with the threat for the time period it is confronted is useless.[34]

As the intended environment of most military equipments is combat, operational testing must of necessity fall short of its goal. It must, however, as a practical matter be conducted.[34] Because the combat environment can only be approximated or simulated and because any sound testing program must be reproducible, the planning of the test looms as a crucial stage in operational testing. In fact, the Operational Test and Evaluation Agency's "Test Methodology" (Volume II, Annex H) states,

> Sound test planning is a major determinant of test relevance, comprehensiveness, and validity. The key aspects of sound test planning are the decisions regarding how the critical issues will be tested, what measures of effectiveness will be used, . . .

Naturally a thorough understanding of the system to be evaluated is preliminary to any test design. The functions, missions, or services provided by the system to a large extent tend to determine the operational issues and their criticality.

Once the critical operational issues are selected, however tentative they may be initially, an effort must be made to understand the factors within the complete system that might contribute to a failure of this issue. Of course thorough testing should reveal most of these factors, but an awareness of their probable presence before testing, can save time and dollars through improved test design.

The word complete in the preceding paragraph should be stressed, as no system stands alone. As stated earlier,

in the context of actual employment in a combat environment or of operational testing of complex military systems, the system interfaces or contains existing sub-systems, personnel, and equipment. Particularly in a command and control system, people, both operating personnel and users, play an important role in system effectiveness. In fact, the principal system undergoing scrutiny inevitably is dependent upon one or more lesser military systems or components. Such systems or equipments which provide a communications capability or a power-generating system are inseparable from the primary system for any measure of overall effectiveness. It is the author's contention that no preliminary planning for test design can be complete unless the impacts of these system and personnel interfaces and relationships within the system under study are fully explored.

Definition of the Problem

The detection of the factors within the complete system that contribute to the failure of an operational issue is the primary thrust of this research. The author is proposing a methodology which will identify or predict those unique combinations of major components, personnel, and sub-systems which can cause failure of an issue and their relative importance to the failure of interest. Additionally, any factors common to several unique modes of failure, can be readily identified and their relative

importance determined.

Brief History Leading to the Problem

Historically operational test and evaluation has been located at the six service test boards assigned to the U.S. Army Continental Army Command (CONARC). In mid-1962, the Army Material Command (AMC) consolidated in the Test and Evaluation Command (TECOM) the separate test facilities of the technical services, used for engineering or developmental tests, and the six CONARC boards. Testing under this configuration represented the developer, the producer, and the user: it was felt this representation of all interested parties was more efficient. Later that same year, the U.S. Army Combat Developments Command was created to represent the user. USA CDC was also responsible for establishing the means for achieving better use of field tests, experiments, and evaluations.[34] During the mid-sixties to early seventies, in response to military concern and congressional criticism, several investigative boards conducted in-depth reviews of test and evaluation within the Army. These boards successively caused: (1) a careful distinction between material testing and operational evaluation, (2) major changes to regulations concerning test and evaluation, and (3) a reorganization creating eventually an agency specially charged with operational test and evaluation, OTEA.

Operational testing then has been at least a concept in the military services for a considerable length of time: its principal short-comings until recently, being a lack of any centralization of responsibility and a lack of sufficiently formalized procedures to insure reproducibility.

In the private sector, on the other hand, operational testing and evaluation is only recently emerging. Corporate producers are profit motivated, whereas defense services are user and performance oriented. This difference of orientation bears heavily on allowable risk. Testing in the automotive industry, for example, is oriented toward product improvement. There is a limited amount of testing done which approximates normal usage. It is primarily intended to insure quality control, to meet specified requirements, or to determine future product improvements. However, recent federal regulations concerning automobile safety and air pollution have caused increased operational testing. The need to determine the best and most economical way to decrease exhaust pollution, the need to determine the effects of collisions, etc. have forced this increase.[34]

Still, the agencies of the Department of Defense appear to be much more advanced in applying the methodologies of operations research and systems analysis to problem solving than the private sector.[34] This emphasis is necessary since the military is not as interested in

product improvement. It emphasizes large increases in operational capabilities which often challenge the state of the art. These sophisticated systems are often quite complex.

Purpose of the Research

There exist a number of factors influencing the military and the author which caused this research. Primarily the need for independent evaluation of reproducible, formal experiments conducted upon highly complex systems places great emphasis upon the test planning or design phase. The relative lag between operational testing in the private sector and in the military sector concentrated the author's attention to current procedures in the military. A personal conviction that a thorough understanding of any system is essential prior to use or to testing, and an admittedly personal bias that all too often communications systems and communicators are forced to respond to established plans, etc. rather than being consulted in the planning phases are among these factors. These factors led to a search for a conceptually simple method which would aid the test planner to more fully understand the complete system undergoing testing and which would highlight the causative or critical factors affecting success.

Fault tree analysis was selected. It is not the intent of this research, however, to significantly extend fault tree theory. Nor, is its intent to offer a cureall

or panacea for operational test design. It is an attempt to apply the existing theory of fault trees and their analysis to an area other than safety and reliability, specifically to test design. This research attempts to demonstrate how the logic of fault trees can be applied by the individual test officer to complex systems in order to insure the completeness of test planning.

Review of the Literature

There exists a wealth of literature covering test planning or test design over a wide range of subjects. However, little has been written concerning operational test design except in governmental reports concerned with specific systems or sub-systems. When one gets even more specific, concerning oneself with the operational test planning for complex or sophisticated systems, there is little to be found of general applicability in use.

One method commonly used in large, complex systems which truly challenge the state of the art and as a consequence are expensive, is termed by some as "test bedding." In "test bedding" a developmental configuration of the system is tested. This equipment may be far from the actual equipment specifications; it can be commercially available equipment similar to the eventual system. The procedure here is to explore different configurations, organizations, etc. with the test bed equipment in an effort to learn more about the system.[20] This procedure does have the

advantage of timeliness, preproduction, but is obviously expensive. It is expensive not only for the equipment, but in terms of the personnel and the time necessary to thoroughly explore the system. Perhaps an excellent example or source of more information on this process of "test bedding" is the testing of automated command and control systems (management information systems adapted for military use) as is currently being done at Fort Hood, Texas. Further, this technique is one normally employed during developmental testing; the results of which are subject to OTEA independent evaluation. However, it is not one available to the individual test officer.

One technique available to the test officer is the dendritic structure technique which is intended to derive the operational issues. Briefly, the method consists of analyzing the functions of the system down to its component parts, stating the required capability for each level, converting the functions and capabilities to issues, and arranging the issues into a tree-like structure.[30] This technique has the effect of increasing understanding of the system. It is not generally evaluated; however, it can be through computer simulation after weights have been assigned to the issues.

Computer simulation is the technique which presently provides those identifications and interrelationships which were discussed earlier. For a detailed discussion of

simulation methods commonly used in military operations research, the reader is referred to the report by W. K. McQuay.[24] An exacting simulation program of a complex system can be expensive, both in time and money.[30] It is quite versatile, however. A simulation program can consider a wide range of system employments; it can provide many economical replications; and it is highly reproducible. There is no intent here to supplant computer simulation as an invaluable aid to test design in large expensive systems. Yet there is the intent to supplement it, to decrease cost, and to increase timeliness. When exploring configurations and employments, in the words of Hamilton and Nance (1969), "one 'wiggles' each part, in order to see what will happen when all the parts are taken into account." The intention here is to isolate those parts that need wiggling when a computer simulation is available or to provide a framework for wiggling on a more economical scale.

An extensive review of the literature concerning fault trees was conducted. This section mentions but a few of the major references in some rather loosely grouped areas of interest. Fault tree analysis was developed by Bell Telephone Laboratories in 1961. In 1965 several papers were presented at the Safety Symposium sponsored by the University of Washington and the Boeing Company.[39] This event marked the beginning of interest in fault tree analysis as a method for determining the safety and/or

reliability of large complex systems, particularly in nuclear reactors and chemical processes. Lambert (1973) and Hassl (1965) give good discussions of fault tree construction. A fine description of fault tree techniques and concepts is given by Fussell (1973). The literature is replete with various methods for quantitative analysis with and without computer applications. Some of the more prominent in this area are Vesely (1970), Fussell (1973), and Crossetti (1969). Appendix A discusses a few of the available computer programs in the field. For a more detailed discussion of some of these the reader is referred to Field et al. (1974). In the area of qualitative analysis, the central thrust of this research, the reader is referred to Chatterjee (1974), Barlow et al. (1974), and Larsen (1974). Fault trees have received only limited attention in areas outside reliability and safety. However, for a brief discussion of other alternatives such as schedule delays in air traffic control systems and the evaluation of market alternatives, the reader is referred to Crossetti (1970).

Organization of the Research

Chapter I introduces the problem, provides the reader with some background on the motivation and history of the problem, and brings the reader up to date on the present practices and literature.

Chapter II introduces the fault tree methodology used to solve the problem, discusses the rationale for

selecting the algorithms used, and demonstrates the use on a small scale problem.

Chapter III introduces a problem to the analytical resolution of the fault tree. It suggests a method which while approximate, increases the efficiency of manual computation.

Chapter IV introduces a typical automated command and control system proposed for use in the U.S. Army. The fault tree methodology is applied to the problem to demonstrate its use.

Chapter V draws some conclusions about the validity of the approach and its limitations, and gives some recommendations for future research.

Appendix A provides a short discussion of quantitative analysis and of the capabilities of some available computer programs.

CHAPTER II

FAULT TREE ANALYSIS

This chapter discusses the theory and application of fault tree analysis. The theoretical basis and the application of the various algorithms are given and applied to a small scale problem. Brief commentaries on the alternative methods with recommendations are included.

General

Fault tree analysis is a technique used for reliability analysis. Its major application has been to highly complex systems. Fault tree analysis is an inclusive, versatile, mathematical tool which provides an objective basis for analyzing system design, performing trade-off studies, identifying common mode failures, and justifying system changes.[10] The fault tree is an evaluation of an undesired or critical event. It utilizes a backward approach working from the undesired event to its causes. It is not restricted to hardware, but can also identify non-hardware causes such as human error and environment.[7]

Some of the major benefits to be gained by the use of fault tree analysis are:

1. The analyst must deductively determine failures.

2. The characteristics of the system bearing on the failure of interest are identified.
3. It provides a visual aid to system understanding to analyst and management alike.
4. Options are available for both qualitative and quantitative analysis.
5. The analyst is permitted to scrutinize one system failure at a time.
6. Lastly, the analyst gains excellent insight into system behavior.[12]

Unfortunately, fault tree analysis is not without disadvantages. These may be mitigated somewhat in the present application. The cost is one cited disadvantage. The cost is a function of both the complexity of the system and the relative scarcity of skilled analysts.[15] However, fault tree analysis is versatile. An example of that versatility is the depth of analysis required. For an operational test design of a military system, it is proposed here to pursue the tree downward only to the end item or major component: this will become clearer in the example which follows later. Done in this manner the systems analyzed are not nearly so complex as those to which fault tree analysis is currently applied. Secondly, as will be seen, fault tree analysis is conceptually not difficult and a test officer trained in fault tree analysis with a background in probability should be able to apply it to most military systems.

Other cited disadvantages of fault tree analysis are: (1) a group of three to five analysts is necessary for systems of moderate complexity; (2) the analysts must become intimately familiar with the system and the basic physics, economics, etc. of the system; and (3) for complex systems, the work can be tedious and time consuming.[15] Again the author feels that (1) and (3) above are compensated for by the relative simplicity of the major component approach. While the understanding described in (2) above is highly desirable in the test officer; thus, it becomes an advantage.

Fault Tree Terminology and Symbology

In essence a fault tree represents a logical statement of the cumulative effect of faults within a system. A system failure of interest is specified: this failure of interest is referred to as the TOP event. This TOP event is then decomposed through a series of logical AND and OR gates and lesser fault events to the primary fault events or basic component faults.[15] To differentiate, at this point, between a basic component fault in fault tree terminology and a major component as used in a description of equipment appears necessary. Take a radio set; major military components would probably be a radio transmitter and a radio receiver as a minimum. However, were one to describe a basic component fault for a fault tree, there could be one

for each fuse, each switch, etc. dependent, of course, on the depth of analysis. There should be no further confusion since equipment components will be among our basic component faults.

Some other basic definitions are in order at this time.

Component Configuration: Description of the component states where the component may have several operating states none of which are necessarily failed.

Fault Event: A failure situation which results from the logical interaction of basic component faults or primary failures.

Branch: The decomposition of any fault event results in a branch of the fault tree.

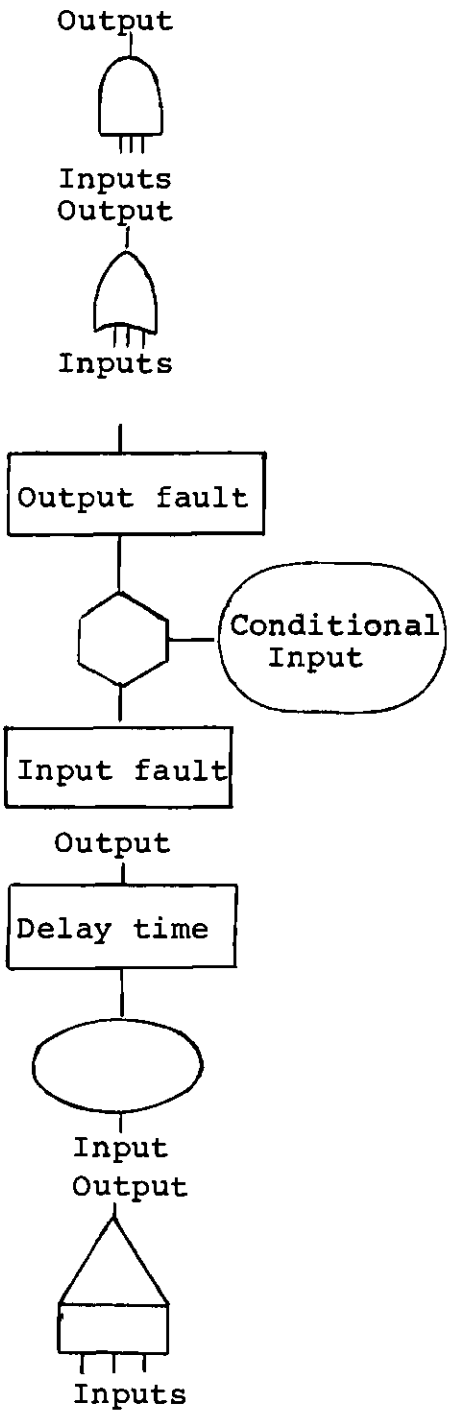
Base Event of the Branch: The fault event which developed leads to the branch.

Domain: Every event in a branch is in the domain of the base event.

Gate: The Boolean logic symbol that shows the action between inputs to the gate and the output.

Minimal Cut Set: The smallest set of primary events which must happen to cause the top event.[10]

The symbols involved in fault tree construction are of two types: logic symbols and event symbols. Logic symbols are shown in Figure 1 and event symbols are shown in Figure 2.[5,32]



AND Gate: Coexistence of all inputs is required to produce output.

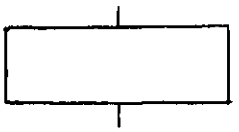
OR Gate: Output will exist if at least one input is present.

INHIBIT Gate: Input produces output directly when conditional input is satisfied.

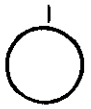
DELAY Gate: Output occurs after specified delay time has elapsed.

Matrix Gate: Output is related to one or more unspecified combinations of undeveloped inputs.

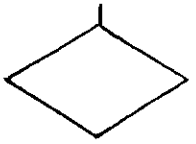
Figure 1. Fault Tree Logic Symbols



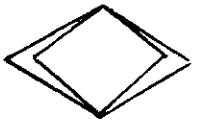
Rectangle: A fault event usually resulting from the combination of more basic faults acting through logic gates.



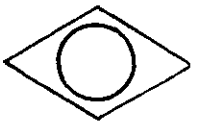
Circle: A basic component fault, an independent event.



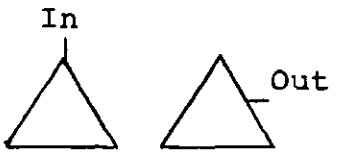
Diamond: A fault event not developed to its cause.



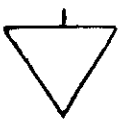
Double Diamond: A significant undeveloped fault event that requires further development to complete the tree.



Circle-Diamond: A fault event, independent of the rest of the tree, was developed separately. Treated as a component.



Triangle: A connecting or transfer symbol.



Upside Down Triangle: A similarity transfer--the input is similar but not identical to the like identified input.

House: An event that is normally expected to occur. Also useful as a "trigger event" for logic structure changes within the tree.

Figure 2. Fault Tree Event Symbols

Logic symbols or gates are used to connect the fault events that can lead to the specified TOP event. The most commonly used gates are the AND gate and the OR gate. The output from an AND gate occurs if and only if all the inputs occur. The output from an OR gate occurs if one or more inputs are present. The circle, diamond, and rectangle are the most frequently used event symbols. The circle represents a basic component fault. The diamond indicates a fault event that is considered basic in the fault tree; however, it is not basic in the system. This status could be a result of the relative inconsequence of the event or a lack of sufficient information about the event. Circles and diamonds generally represent primary events which are mutually independent. The rectangle represents a fault event which is a function of more basic fault events acting through logic gates. They are often considered as a part of their associated logic gate.[13]

System Definition

Before the first step in fault tree construction is taken the system must be defined. An important source of information here would be some type of functional layout diagram. This diagram should show all functional interconnections and identify all components. For some systems that are not hardware oriented, a functional diagram may not exist: the fault tree could be the best diagram

available.[11] Most systems in an operational test environment will have some associated hardware and of course some human users and operators. The latter are generally not reflected in a functional diagram: the author suggests, insofar as possible, they be added to the diagram for clarity.

Bounds for the physical system must then be established. An approach to this would be to state the functional purpose of the system in words being as specific as possible. This statement should contain the what, when, and where of the system.[22]

Next, one progresses to component identification. Begin this step by identifying sub-systems and then identify the components within each sub-system.[22]

The last step in system definition is to specify the system boundary conditions. Do not confuse these system boundary conditions with the physical bounds. The physical bounds are constraints on the system which generally do not change. The system boundary conditions are situation specific. That is, they define the conditions for which the fault tree is to be drawn. The most important system boundary condition is the TOP event. For any system many TOP events, major system failures of interest, exist. The initial configuration of the system, which must be an unfailed configuration, is specified by additional system boundary conditions. Special attention must be

given at this point to those components that have more than one operating state. Further system boundary conditions are necessary to specify those fault events always existing or not allowed to exist for the purposes of the fault tree. Such events are termed existing boundary conditions or not-allowed boundary conditions.[10]

An example now will help to assimilate these concepts. Take a simple anti-tank or similar weapon, SIMGUN. SIMGUN consists of a firing device, gun tube, an explosive projectile with a homing device, a control panel allowing manual override if the homer should fail, a power source for the control panel, and an operator. A simple schematic is shown in Figure 3.

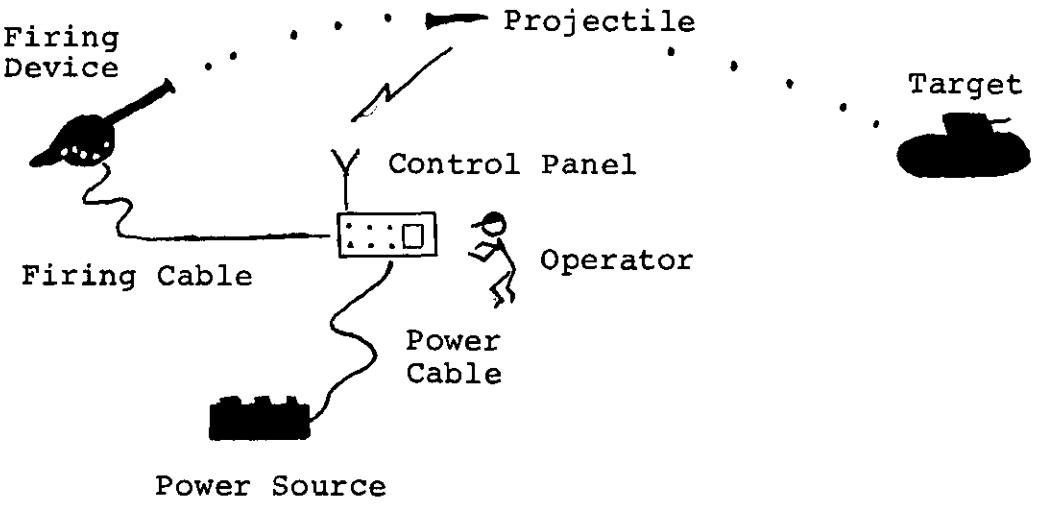


Figure 3. SIMGUN Schematic

Functional Purpose:

What: Eliminate or disable a moving or static armored vehicle at a range of 5,000 meters with a 95 percent probability and a 90 percent hit probability.

When: 5 minutes allowed from target sighting to hit.

Where: in a combat environment.

These system bounds can be as elaborate as necessary. As a minimum those characteristics that may affect the fault tree analysis should be specified.

Sub-systems and Components:

Projectile: propulsion device
homing device
radio element

Firing device: firing device

Control panel: control panel
power supply

Operator: operator

System Boundary Conditions:

TOP Event: miss target

Initial condition: System checks operable
No component has more than one operating state

Not-allowed events: Cable failures
Failures due to effects external to system

Existing effects: None

Fault Tree Construction

There is little information in the published literature about general fault tree construction. Fussell (1972) has presented a formal methodology for electrical systems and Haasl (1965) has described some general concepts. Lambert (1973) has established some rules for fault tree construction which are repeated here.

Rule 1: State the fault event as a fault, including the description and timing of a fault condition at some particular time. Include:

(a) What the fault state of the system or component is.

(b) When that system or component is in the fault state.

Test the fault event by asking:

(c) Is it a fault?

(d) Is the what and when portion included in the fault statement?

Rule 2: There are two basic types of fault statements: state-of-system and state-of-component.

To continue the tree

(a) If the fault statement is a state-of-system statement, use Rule 3.

(b) If the fault statement is a state-of-component statement, use Rule 4.

Rule 3: A state-of-system fault may use an AND,

OR, or Inhibit gate or no gate at all. To determine which gate to use, the faults must be the

(a) Minimum necessary and sufficient fault events.

(b) Immediate fault events.

To continue, state the fault events input into the appropriate gate.

Rule 4: A state-of-component gate always uses an OR gate. To continue, look for the primary, secondary, and command failure fault events. Then state those fault events.

(a) Primary failure is failure of that component within the design envelope or environment.

(b) Secondary failures are failures of that component due to excessive environments exceeding the design environment.

(c) Command faults are inadvertent operation of the component because of a failure of a control element.

Rule 5: No gate-to-gate relationships.

Rule 6: Expect no miracles; those things that would normally occur as a result of a fault will occur, and only those things. Also normal system operation may be expected to occur when faults occur.

Rule 7: In an OR gate, the input does not cause output. If any input exists, the output exists. Fault events under the gate may be restatement of the output

events.

Rule 8: An AND gate describes a causal relationship. If the input events coexist, the output is produced.

Rule 9: An inhibit gate describes a causal relationship between one fault and another, but the indicated condition must be present. The fault is the direct and sole cause of the output when that specified condition is present. Inhibit conditions may be faults or situations, which is why AND and inhibit gates differ.[22]

A comment on Rule 4 is needed. These rules are designed for safety analysis. Hence, as used in this research secondary failures are generally not allowed. Command faults are often considered by a component termed operator error.

A fault tree for the SIMGUN weapons system is shown in Figure 4. It is "a" fault tree rather than "the" fault tree for two reasons: (1) each system can have many TOP events, and (2) two analysts working on the same system with the same top event will seldom create identical trees. As a convention the author uses the diamond to indicate the human component.

Qualitative Analysis

Minimal Cut Set Algorithms

A cut set of a fault tree is a set of inputs, primary fault events, which together cause system failure. A

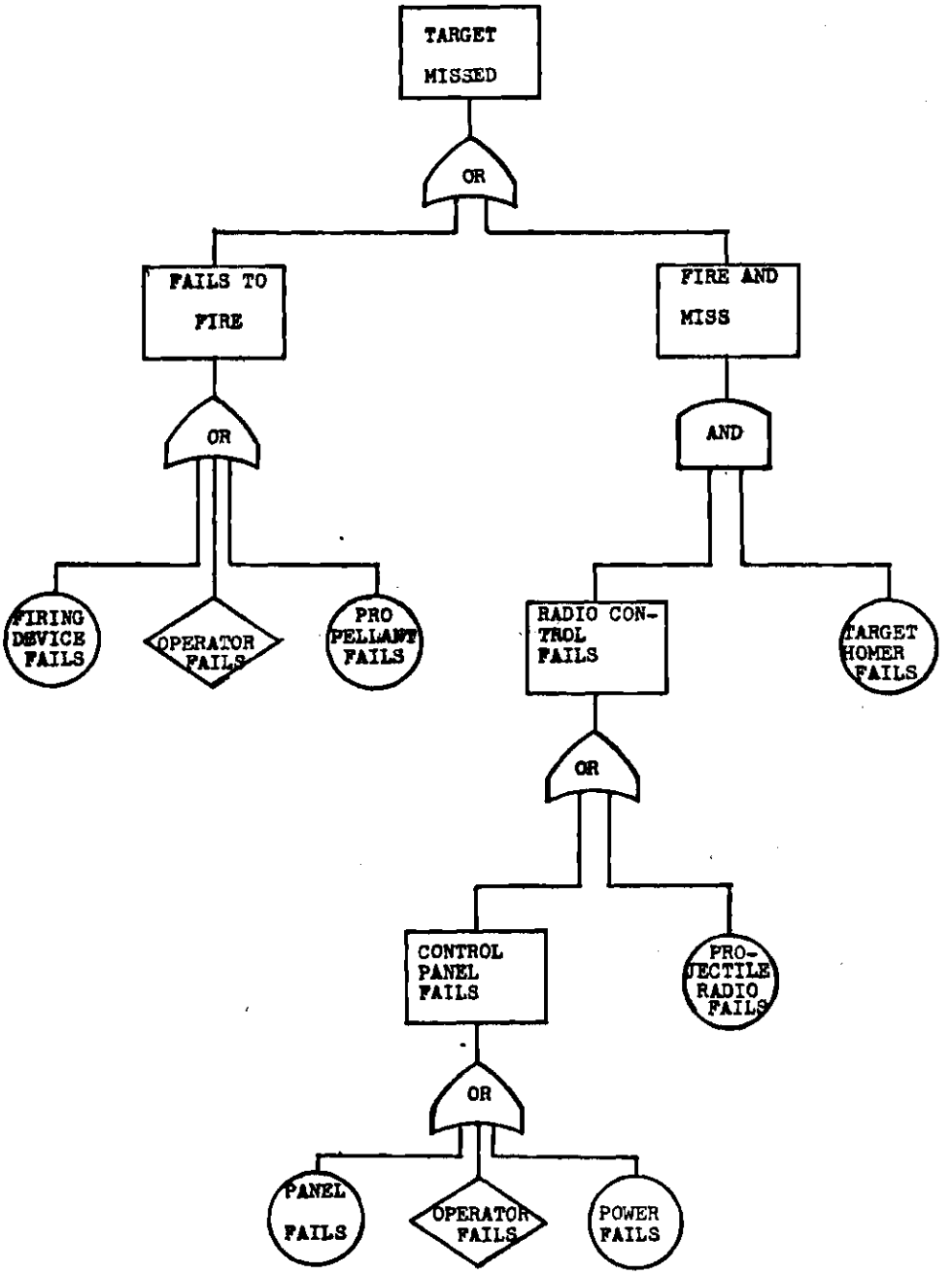


Figure 4. SIMGUN Fault Tree

minimal cut set is a cut set such that no strict sub-set is also a cut set.[36] For example, let (1,2,3) and (1,3) be cut sets of a fault tree where 1,2,3 represent basic component failures. The set (1,2,3) is not a minimal cut set since the failure of components 1 and 3 is sufficient to cause system failure. The determination of the minimal cut sets is a major step in fault tree analysis.

In some of the Army systems, analyzed to the depth discussed earlier, the minimal cut sets can be determined by inspection. For instance, for the coded SIMGUN fault tree shown in Figure 6, the minimal cut sets are: B,H,C, FD,GD, and ED. However, inspection is hardly the best means because fault trees for larger systems would prohibit this technique.

The first technique to be discussed is the Boolean reduction method. This is a process of manipulating the coded tree according to basic Boolean operations. Consider the gates shown below in Figure 5. The AND gate shown in

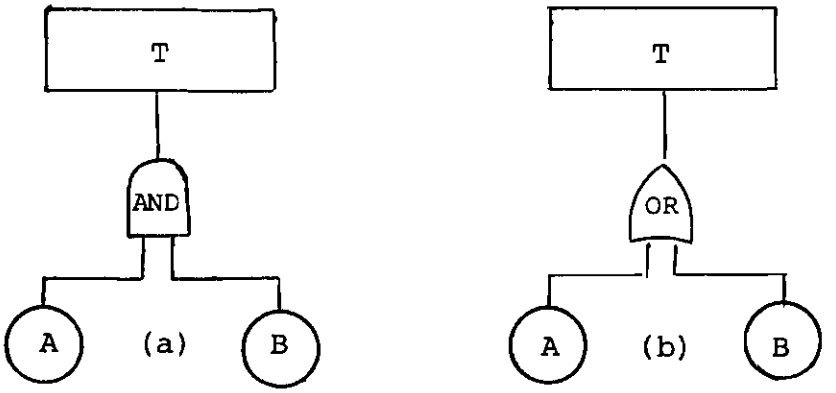


Figure 5. AND and OR Gates

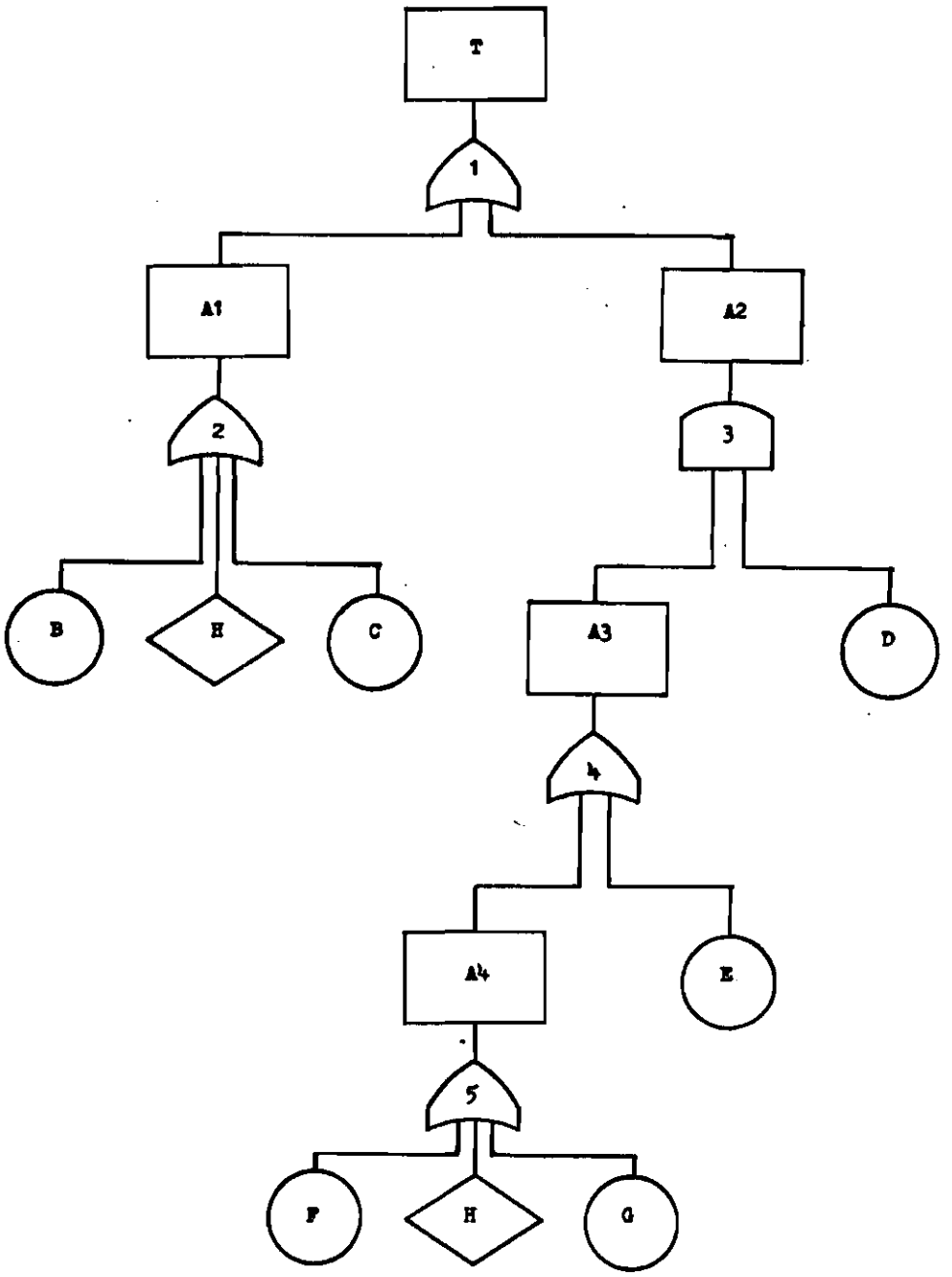


Figure 6. Coded SIMGUN Fault Tree

5a is equivalent to the Boolean expression: $T = A \cap B = AB$. The OR gate in 5b is equivalent to $T = A \cup B = A+B$. Boolean algebra is an algebra of sets. Various laws and theorems are summarized below.[17] $\overline{A}$ is the logical complement of A.

Commutative laws: $AB = BA$

$A + B = B + A$

Associative laws: $A(BC) = (BC)A$

$A + (B+C) = (A+B) + C$

Distributive laws: $A(B+C) = AB + AC$

$A + BC = (A+B)(A+C)$

DeMorgan's laws: $\overline{(AB)} = \overline{A} + \overline{B}$

$\overline{(A + B)} = \overline{A}\,\overline{B}$

Other laws:

$0\,A = 0$	$A + A = A$
$1 + A = 1$	$A(A + B) = A$
$1A = A$	$A + AB = A$
$0 + A = A$	$A\,\overline{A} = 0$
$\overline{0} = 1$	$A + \overline{A} = 1$
$\overline{1} = 0$	$\overline{(\overline{A})} = A$
$A\,A = A$	

Returning to the SIMGUN example, the Boolean equations are:

$$T = A1 + A2 \qquad A3 = A4 + E$$

$$A1 = B + H + C \qquad A4 = F + H + G$$

$$A2 = A3(D)$$

Substituting yields

$$T = B + H + C + (F + H + G + E)D$$

Removing redundancies according to laws of Boolean algebra, the expression $T = B + H + C + FD + GD + ED$ is obtained. The terms on the right side of the equality correspond exactly to the minimal cut sets. In general there is a one to one correspondence between the minimal cut sets for a fault tree and the terms of the fully expanded, non-redundant Boolean expression of the TOP event.[36]

The second method has probably received the most recognition; it was introduced under the name MOCUS by Fussell and Veseley (1972). It is a downward moving algorithm. The basis for this algorithm is that an AND gate always increases the size of a cut set while an OR gate always increases the number of cut sets. If all the primary events are different, this algorithm will immediately give all the minimal cut sets. If replication occurs among the primary events, a search will reveal the minimal cut sets.[10] An example is the simplest way to explain the use of this algorithm: refer again to the SIMGUN example. Create a list matrix using the inputs to the OR gate under the TOP event.

A1

A2

An OR gate creates additional rows; an AND gate, additional columns. Now replace A1.

B

H

C

A2

Now, A2 which is preceded by an AND gate.

B

H

C

A3 D

Now, A3.

B

H

C

A4 D

E D

Now, A4.

B

H

C

F D

H D

G D

E D

Since primary event H was replicated in the tree; a search reveals that H D is not a minimal cut set, the minimal cut sets are B, H, C, FD, GD, ED which agrees with previous results.

The last algorithm for determining minimal cut sets is an upward moving algorithm, starting from the bottom of the tree. MICSUP, the algorithm, was developed by Chatterjee (1974). This time starting at the bottom of the tree, one obtains G_i, the set of all cut sets inputted to gate "i". The minimal cut sets are determined by a simple search to obtain G_i^*. For more complex systems, more effective methods of searching are discussed by Chatterjee (1974). This reduction is unnecessary when the set of basic events inputted are mutually disjoint. For the case of an OR gate G_i is the union of all inputted G_j^*. For the case of an AND gate, G_i is the cartesian product.

The reader is referred to the SIMGUN example.

$$G_5 = G_5^* = (F,H,G)$$

$$G_4 = E \cup G_5^* = G_4^* = (E,F,H,G)$$

$$G_3 = D \times G_4^* = G_3^* = (DE,DF,DH,DG)$$

$$G_2 = G_2^* = (B,H,C)$$

$$G_1 = G_2^* \cup G_3^* = (B,H,C,DE,DF,DH,DG)$$

Reduction is necessary since H is replicated: the inputs are not mutually disjoint. Hence, the minimal cut sets are:

$$G_1^* = (B,H,C,DE,DF,DG)$$

which again agrees with previous results.

Of the three algorithms presented, the author recommends the last, MICSUP, the upward moving algorithm, when used in test design. It provides the minimal cut sets of any intermediate event of interest. This can be very advantageous when looking at several TOP events for the same system. Sub-systems often repeat themselves with an identical tree structure for different TOP events on the same system. Additionally, when computerized, solving a tree that exceeds storage requirements; the first run is not wasted. This identification of minimal cut sets for intermediate events is not possible with the other two algorithms; consequently, in computer runs with the downward algorithm which exceed storage capacity, as many computer runs as there are unexpanded gates are necessary.[3]

Computer execution time with the upward algorithm is generally small compared to the downward algorithm.[3] In the use presented here, manual computation with the first two algorithms is quite tedious. This circumstance occurs with replication among primary events and the necessity to carry all terms downward until the replication allows reduction. In the upward algorithm the terms are reduced as they enter the tree. A situation having parallel redundancy of sub-systems is particularly awkward. Parallel redundancy is common enough in any critical

system, but this combination of several OR gates inputting to an AND gate is very cumbersome, particularly with the first two algorithms presented.

Minimal Cut Set Importance

The importance of minimal cut sets, discussed in this section, and the importance of components, discussed in the next section, are not generally included under qualitative analysis. In fact, qualitative analysis is generally restricted to identification of the minimal cut sets and the ordering of those cut sets by size. These features are included under qualitative analysis here to highlight the fact that they are not dependent upon probabilistic failure data for the primary events.

In the literature search of fault tree analysis, only two methods of determining minimal cut set importance were discovered.

Fussell (1974) presents one method of determining minimal cut set importance. I_k is defined as the probability the mode failure is causing system failure when the system has failed. Let A be defined as the event a minimal cut set K has failed and B as the event the system is in the failed state, i.e., the TOP event has occurred. Then,

$$I_k = P(A|B) = \frac{P(A)\ P(B|A)}{P(B)}$$

However, given that a minimal cut set has failed the

probability the system has failed is $P(B|A) = 1$. Hence,

$$I_k = \frac{P(A)}{P(B)} \tag{1}$$

Those minimal cut sets with the highest value of I_k are the most critical.[11]

Assume all component features equally likely and equal to 0.05, i.e., $P(B) = P(H) = \ldots = P(G) = 0.05$. This methodology is applied to the SIMGUN example.

A simple approximation for P(B), the probability of the TOP event, is obtained by substituting directly into the Boolean equation for the TOP event. This greatly simplifies the calculations and will give a more pessimistic result.[23] Hence,

$$P(B) = P(B) + P(H) + P(C) + P(DE) + P(DF) + P(DG)$$

$$P(B) = .05 + .05 + .05 + .0025 + .0025 + .0025 = .1575$$

Obviously then,

$$I_B = I_H = I_C = .32$$

and

$$I_{DE} = I_{DF} = I_{DG} = 0.02$$

These results are not startling: one would expect all minimal cut sets consisting of one component to be more important than those consisting of two components. The weakness here is that some subjective estimates of the

various component probabilities of failure have to be made; any assumption of equality or lack of knowledge yields trivial results.

The second method presented is due to Barlow and Proschan (1974). Letting the probability of failure of all components be equally likely, the structural importance is:

$$I_K = \sum_{i \varepsilon K} \int_0^1 h(1_i, \underline{0}^{K-\{i\}}, p)(1-p)^{k-1}\, dp \qquad (2)$$

where k is the number of components in minimal cut set K, and "i" indicates components. $h(1_i, \underline{0}^{K-\{i\}}, p)$ is the probability that component "i" is critical: it represents a component operational status where all components in the minimal cut set are not operational (=0) except i which is operational (=1) and all other components are operational with a probability p. $(1-p)^{k-1}$ is, of course, the probability that the remaining components in K have failed. The product of these two terms yields the probability that component i causes failure. Integrating over p is equivalent to assuming p is distributed uniformly on [0,1] and summing over $i \varepsilon K$ corresponds to the mutually exclusive ways in which cut set K can fail.[2] If the preceding heuristic explanation of the equation is not sufficient, the reader is referred to Barlow and Proschan (1974) for the theoretical development and proof.

The problem in the application of this method to

operational testing is the determination of $h(1_i, \underline{0}^{K-\{i\}}, p)$, the probability that component i is critical. By critical is meant that if i is operating the system is operating, when it fails the system fails.

For reasons that will become clear shortly, a dual fault tree is constructed in Figure 7 for the SIMGUN example. A dual fault tree is simply a tree for the non-occurrence of the TOP event. To draw the dual fault tree simply replace each OR gate with an AND gate, each AND gate with an OR gate, and each event with its corresponding dual. The minimal cut sets for the dual fault tree are the minimal path sets for the original fault tree. A minimal path set is a non-reducible set of basic events whose nonoccurrence insures the nonoccurrence of the TOP event. From Figure 7 the minimal path sets are (BHCD) and (BHCEFG).

Hence, the Boolean expression for the probability of the nonoccurrence of the TOP event is:

(NOT MISSING TARGET) + (BHCD) + (BHCEFG)

This Boolean expression in our example is not truly an expression for h(.); it is at best an approximation. If the sets (BHCD) and (BHCEFG) were mutually exclusive, it would be. However, mutually exclusive events are not to be expected at the level of analysis here, but the reader is referred to Fussell (1973) for the treatment. In fact,

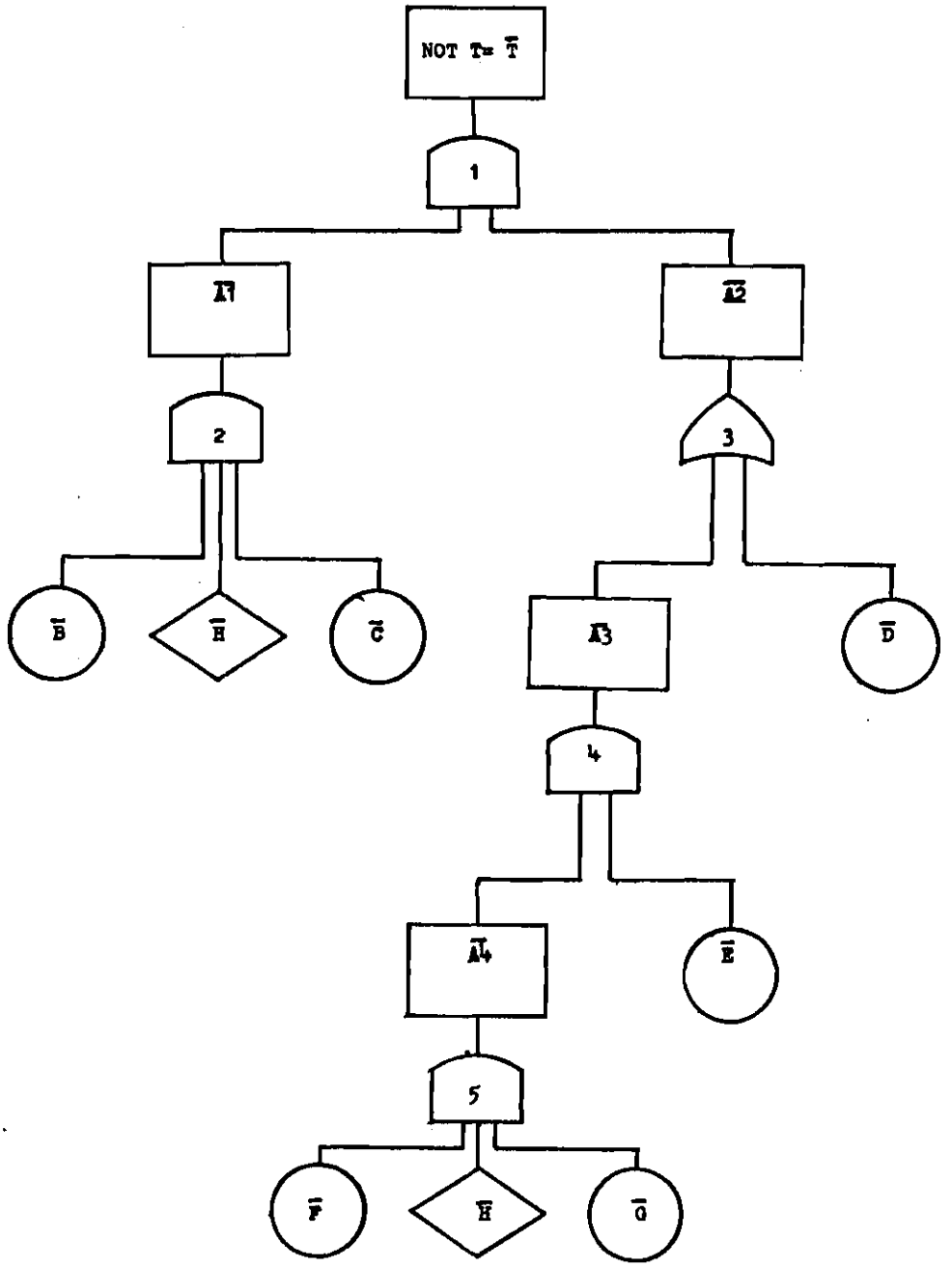


Figure 7. Dual SIMGUN Fault Tree

converting the Boolean expression to the equivalent probability expression yields,

$$(\overline{T}) = (BHCD) + (BHCEFG) - (BHCD)(BHCEFG)$$

Recall that (BHCD)(BHCEFG) is an intersection, and that (BHCD) and (BHCDFG) are intersections. BB = B, hence,

$$(BHCD)(BHCEFG) = BHCDEFG.$$

Now putting this expression into the expression for I_H, i.e., letting H = 1 and all other components = p since H is a one-element cut set. One obtains:

$$I_H = \int_0^1 (p^3 + p^5 - p^6)(1-p)^0 dp$$

$$= \int_0^1 p^3 + p^5 - p^5 dp$$

$$= \frac{1}{4} + \frac{1}{6} - \frac{1}{7} = .274$$

and

$$I_{DE} = \int_0^1 p^3(1-p)dp + \int_0^1 p^5(1-p)dp$$

$$= \frac{1}{4} - \frac{1}{5} + \frac{1}{6} - \frac{1}{7} = .074$$

In a similar manner, it can be determined that

$$I_B = I_C = I_H = .274$$

$$I_{DG} = I_{DF} = I_{DE} = .074$$

Again the results are not startling; the ordering is as expected. However, with the first method results were identical for cut sets of the same size due to the selection of equal probabilities. For this method, it is a quirk of the sample problem. It can be readily seen that this determination of importance is a function of both the size of the cut set and the number of minimal cut paths in which each component is contained.

Note the summation of the importances is greater than unity, $\Sigma I_K \geq 1$. This is reasonable because some sequences of component failures can cause simultaneous minimal cut set failures. In SIMGUN, if G fails, then F, then D, two cut sets cause failure simultaneously, (GD) and (FD).

Component Importance

The emphasis in this section will be on presenting the methods for determining component or primary event importance.

Using a similar probabilistic derivation to that presented for the importance of minimal cut sets, Fussell (1974) develops an approximation for I(i), the importance of component i,

$$I(i) = \sum_{j=1}^{n} I_j \qquad (3)$$

where n is the number of minimal cut sets containing i and I_j is the importance of the jth minimal cut set containing i.[11]

Applied to the SIMGUN example problem, the following results are obtained.

$$I(B) = I_B = .32 = I(H) = I(C)$$

$$I(E) = I_{DE} = .02 = I(F) = I(G)$$

$$I(D) = I_{DE} + I_{DF} + I_{DG} = (.02)3 = .06$$

However, since these results are a function of minimal cut set importance, the same criticism, the need for a subjective assessment of probabilities, is still valid.

Larsen (1974) presents a rather simplistic procedure to identify those components which have the most influence on the output fault. In other words, it is another method of determining the relative structural importance of components. The steps to be taken in his analysis are as follows:

1. Write the Boolean expression for a failure of the TOP event.

2. Substitute the probability value of 0.1 for each primary event into the Boolean expression and solve.
3. Select a higher probability value, say 0.5, and substitute this value for one input event, holding all others at 0.1. Do this for each primary event.
4. Arrange the results in tabular form in descending order.
5. Divide the new output fault values by the results of step 2. The result is termed the sensitivity rating. The sensitivity rating has no intrinsic value, except to show the relative influence of each component on the output fault.[23]

This method can now be applied to the SIMGUN example:

(1) P = B + C + H + DE + DF + DG

(2) P = .1 + .1 + .1 + .01 + .01 + .01 = .33

(3) B = .5 + .1 + .1 + .01 + .01 + .01 = .73

C = .73

H = .73

D = .1 + .1 + .1 + .05 + .05 + .05 = .45

E = .1 + .1 + .1 + .05 + .01 + .01 = .37

F = .37

G = .37

(4) Event	Sensitivity rating
B,C,H	2.21
D	1.36
E,F,G	1.12

Birnbaum (1969) proposed:

$$B(i) = \frac{\partial h(p)}{\partial p_1}\Big|_{p_1 = p_2 \cdots p_n = \frac{1}{2}} \tag{4}$$

as a measure of the importance of basic event in a coherent structure. Barlow and Chatterjee (1973) adapted this method to fault trees. They derived an expression for B(i) in terms of n(i), the number of critical path sets for i and n the number of basic events in the tree.[1]

$$B(i) = 2^{-(n-1)}n(i)$$

The number of critical path sets, however, can be a laborious tabulation exercise except in some rather straightforward combinatoric situations. A critical path set should not be confused with the minimal path sets discussed previously. Critical path sets are combinations of all components taken one at a time, two at a time, etc. such that it may contain several minimal path sets each of which must contain i. Consequently, the probabilistic expression for the nonoccurrence of the TOP event is used here for $h(\underline{p})$ in the original equation.

Refer again to the SIMGUN sample problem.

$$h(\underline{p}) = BHCD + BHCEFG - (BHCD)(BHCEFG)$$

then,

$$B(H) = \left.\frac{\partial h(\underline{p})}{\partial p_H}\right|_{p=\frac{1}{2}} = \left[BCD + BCEFG - BCDEFG\right]\Big|_{p=\frac{1}{2}}$$

$$B(H) = (\tfrac{1}{2})^3 + (\tfrac{1}{2})^5 - (\tfrac{1}{2})^6 = .141$$

$$B(B) = B(C) = B(H) = .141$$

$$B(D) = \left.\frac{\partial h(\underline{p})}{\partial p_D}\right|_{p=\frac{1}{2}} = (\tfrac{1}{2})^3 - (\tfrac{1}{2})^6 = .109$$

$$B(E) = B(F) = B(G) = (\tfrac{1}{2})^5 - (\tfrac{1}{2})^6 = .016$$

The last method to be presented eliminates the arbitrary estimation of probabilities, upon which the first three depend. It corresponds to the last method presented for a determination of minimal cut set importance, and can also be found in Barlow and Proschan (1974). In its final form it also is a function of the number of critical path sets of size r, which again the author rejects for the reasons cited in the section on minimal cut set importance.

Instead one works with the basic definition of component structural importance,

$$I(i) = \int_0^1 [h(1_i,p) - h(0_i,p)]dp \qquad (5)$$

where $(1_i,p)$ is the vector having a 1 in the ith position

and a p in all others. $[h(1_i,p) - h(0_i,p)]$ represents the probability that the system is functioning if i is functioning, but is failed if i is not functioning (=0).[2]

As opposed to the importance of minimal cut sets, the importance of components, I(i), do sum to unity;

$$\sum_{i=1}^{n} I(i) = 1$$

and,

$$0 \leq I(i)$$

Letting $h(\cdot)$ be represented by the equivalent probability expression for the nonoccurrence of the TOP event

$$h(\cdot) = \overline{T} = BHCD + BHCEFG - (BHCD)(BHCEFG),$$

one can solve the SIMGUN sample problem using this definition of component importance.

$$I(H) = \int_0^1 [(p^3 + p^5 - p^6) - (0)]dp$$

$$= \frac{p^4}{4} + \frac{p^6}{6} - \frac{p^7}{7}\Big|_0^1 = \frac{1}{4} + \frac{1}{6} - \frac{1}{7} = .274$$

Again,

$$I(B) = I(C) = I(H) = .274$$

$$I(D) = \int_0^1 [(p^3 + p^6 - p^6) - (p^6)]dp$$

$$= \frac{p^4}{4} - \frac{p^7}{7}\Big|_0^1 = .107$$

$$I(E) = \int_0^1 [(p^4 + p^5 - p^6) - (p^4)]dp$$

$$= \frac{p^5}{5} + \frac{p^6}{6} - \frac{p^7}{7} - \frac{p^5}{5}\Big|_0^1 = .024$$

In the same way

$$I(E) = I(F) = I(G) = .024$$

And,

$$\sum_{i=1}^{n} I(i) = 1 .$$

In the absence of good probabilistic information regarding the basic components, the methods of Barlow and Proschan are to be preferred for determining component importance and minimal cut set importance. If good probabilistic data are available--an unlikely event in the early stages of test design--the reader is referred to Appendix A for a discussion of quantitative evaluation and computer programs.

CHAPTER III

A DILEMMA

The intent of this research is to present a method of analysis not overly complex nor time consuming. In the SIMGUN example used in the preceding chapter, the reader may have missed a serious drawback to manual resolution of minimal cut set importance or component importance. Consider a system consisting of four minimal path sets: A, B, C, D. The Boolean expression for the nonoccurrence of the TOP event, $\overline{T}$, is,

$$\overline{T} = A + B + C + D$$

Conversion to an equivalent probability statement yields,

$$P_{\overline{T}} = p_a + p_b + p_c + p_d - p_a p_b - p_c p_d - p_a p_c \qquad (6)$$

$$- p_a p_d - p_b p_c - p_b p_d + p_a p_b p_c + p_b p_c p_d$$

$$+ p_a p_c p_d + p_a p_b p_d - p_a p_b p_c p_d$$

Computation is no longer simple. When one considers that A, B, C, D each represent minimal path sets which even in the sample problem presented the intersections of four and six component probabilities respectively, it is even less simple. The minimal cut sets will be even larger and

more numerous.

Recall that using the Boolean expression to estimate the probability of failure of the TOP event is acceptable for two reasons:

1. The error caused by neglecting the intersection of two or more small probabilities is slight.

$$(.1)\ (.1) = .01 \qquad (.1)^3 = .001$$

2. The slight increase in the probability of failure caused by this error yields a pessimistic or conservative result.

However, using the Boolean expression obtained from the dual fault tree to represent the probability of success, nonoccurrence of the TOP event, does not appear very reasonable for the same reasons. The error is not as slight for the larger probabilities of successful operation. The result is not conservative. It is extremely optimistic since resolution easily yields probabilities greater than unity.

Fussell (1973) presents a method of Boolean manipulation and substitution which creates an equivalent fault tree in which all failure events are independent. This method is used for the resolution of fault trees which have primary events replicated in different branches of the tree. But replication and dependency become more severe in the dual fault tree. As a result his method,

which is exact, does not simplify calculations.

To further demonstrate the severity of the problem, take the example given at the beginning of the chapter with minimal path sets: A, B, C, and D. Let minimal path set A consist of the intersection of seven components, say (efghijk), and

B = (efhlmn)

C = (emop)

D = (eflk)

The dual fault tree yielding these minimal path sets is shown in Figure 8. Substitute the values for A, B, C, D into the fully expanded probabilistic expression, equation (6). Then,

$$
\begin{aligned}
h(\cdot) = {} & (efghijk) + (efklmop) + (emop) + (eflk) \\
& - (efghijklmn) - (efklmop) - (efghijkmop) \\
& - (efghijkl) - (efhlmnop) - (efhklmn) \\
& + (efghijklmnop) + (efhklmnop) + (efghijklmop) \\
& + (efghijklmn) - (efghijklmnop).
\end{aligned}
$$

Then using equation (5),

$$
\begin{aligned}
I(e) &= \int_0^1 [h(1_e,p) - h(0_e,p)]dp \\
&= \int_0^1 (p^{10} - p^9 + p^8 - 2p^7 - p^6 + p^5 + 2p^3)dp
\end{aligned}
$$

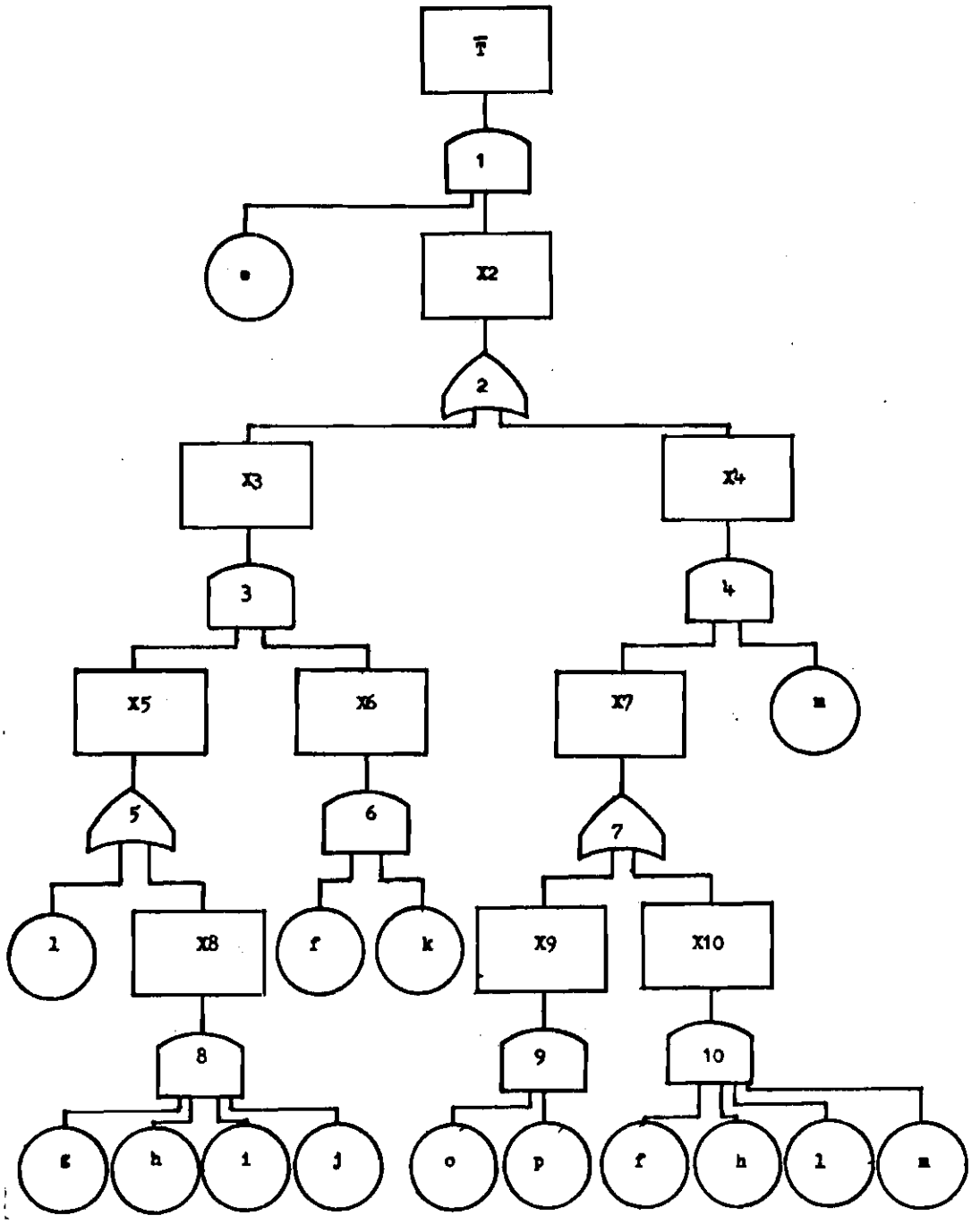


Figure 8. Dual Fault Tree

$$= \frac{1}{11} - \frac{1}{10} + \frac{1}{9} - \frac{1}{4} - \frac{1}{7} + \frac{1}{6} + \frac{1}{2} = .376$$

In a similar manner, determine that

$$I(f) = .126$$

$$I(g) = I(i) = I(j) = .009$$

$$I(h) = .019$$

$$I(k) = I(o) = I(p) = .084$$

$$I(l) = .083$$

$$I(m) = .108$$

$$I(n) = .010$$

The correct component ordering of importance then is:

$$e > f > m > o \sim p \sim k > l > h > n > g \sim i \sim j$$

Clearly the calculations are becoming cumbersome. For more complex systems, a more efficient method is needed.

Resolution for Component Importance

Larsen's method is not complex since it uses the pessimistic Boolean expression for failure. In order to use his method one needs the minimal cut sets. Working from the dual fault tree and making the mental conversion from OR gate to AND gate and visa versa, one obtains the minimal cut sets,

e

fm

fo

fp

km

koh

kol

kon

kph

kpl

kpn

lgm

lgo

lhm

lho

lim

lio

lip

ljm

ljo

ljp

lhp

lgp

The Boolean expression for T is the sum of these terms. Substituting 0.1 for all events one obtains,

$$T = .1 + (.01)4 + (.001)18 = .158$$

Setting each event = 0.5; one at a time.

$e = .5 + (.01)4 + (.001)18 = .558$

$f = .1 + (.05)3 + .01 + (.001)18 = .278$

$g = .1 + (.01)4 + 3(.005) + 15(.001) = .170$

$h = .178$

$i = .170$

$j = .170$

$k = .222$

$l = .214$

$m = .254$

$n = .164$

$o = .226$

$p = .226$

Arranging the events in order yields,

			Sensitivity Rating
e	.558 ÷ .158	=	3.53
f	.278 "	=	1.76
m	.254 "	=	1.61
o,p	.226 "	=	1.43
k	.222 "	=	1.40
l	.214 "	=	1.35
h	.178 "	=	1.13
g,i,j	.170 "	=	1.08

			Sensitivity Rating
n	.164 ÷ .158	=	1.04

This is an ordering which agrees well with the more laborious form earlier. While not that exacting it obtains good results quickly for larger trees. Nevertheless, it is an approximation and as such is not as desirable as the use of equation (5) when time and the problem permit.

The Problem with Cut Set Importance

The problem confronting the determination of component importance exists also in the determination of minimal cut set importance.

Using equation (2) to determine minimal cut set importance on the fully expanded probabilistic expression of this problem one obtains for the first nine cut sets,

$$I_e = \int_0^1 (p^{10} - p^9 + p^8 - 2p^7 - p^6 + p^5 + 2p^3)(1-p)^0 dp = .376$$

$$I_{fo} = \int_0^1 (p^6 + p^5 + p^3 - p^7 - p^6)(1-p)dp + \int_0^1 p^3(1-p)dp = .110$$

$$I_{fm} = .104$$

$$I_{fp} = .110$$

$$I_{km} = .114$$

$$I_{koh} = .039$$

$$I_{kol} = .026$$

$$I_{kon} = .040$$

$$I_{kph} = .039$$

Thus the ordering of these minimal cut sets is,

e > km > of ~ fp > fm > kon > koh ~ kph > kol

Again the computations are tedious; however, for this situation there exists no simpler method. If subjective probability estimates are available Fussell's method for determining minimal cut set importance can be used.

A Proposal

The author offers an extension of Larsen's method to determine minimal cut set importance. The analyst must first assume that all minimal cut sets of size 1 are more important than those of size 2, size 2 more important than size 3, etc. This assumption has intuitive appeal and is generally accepted by most analysts. However, examples may be found where a cut set of size r-1, is more important than several of size r. These examples are functions of the absence of the components contained in r-1, from all

other cut sets and a high degree of component replication among the cut sets of size r.[2] So the minimal cut sets should be scrutinized to determine if this situation exists. Look at the larger cut sets closely to determine if any consist of components which are not replicated in the smaller cut sets.

The minimal cut sets of each size are then evaluated in a manner similar to Larsen components. Let S(K) be defined as the sensitivity rating of cut set K.

$$S(K) = \sum_{i \in K} \frac{T(.5_i, 1^{K-\{i\}}, \underline{.1})}{T(1^{K-\{i\}}, \underline{.1})} \qquad (7)$$

where $T(.5_i, 1^{K-\{i\}}, \underline{.1})$ represents the Boolean expression for failure with .5 in the ith position, 1 in the K-{i} positions, and .1 for all other components. The quotient for each i then represents the sensitivity rating of i when the other components in the minimal cut set are already failed. While summing over $i \in K$ considers the various ways in which K can fail. As in Larsen's method of determining component importance the numbers obtained are intrinsically worthless, except to indicate the relative ordering of the minimal cut sets.

Returning to the example, equation (7) yields,

$$S(e) = \sum_{e} \frac{T(.5_e, \underline{.1})}{T(1^{K-i}, \underline{.1})} = \frac{.5 + (.01)4 + (.001)18}{.1 + (.01)4 + (.001)18} = 3.53$$

which was included only to demonstrate that S(i) for a minimal cut set of size 1 is identical to the sensitivity ratio for determining component importance. For size 2 minimal cut sets,

$$S(fm) = \frac{[.1 + .5 + (.05)2 + .1 + 4(.01) + 14(.001)]}{[.1 + (.1)2 + (.01)2 + 4(.01) + 14(.001)]}$$

$$+ \frac{[.1 + .5 + (.1)2 + (.05) + (.005)4 + 14(.001)]}{[.1 + (.1)3 + (.01) + 18(.001)]}$$

$$= \frac{.854}{.374} + \frac{.884}{.428} = 4.35$$

In a similar manner,

$$S(km) = \frac{.789}{.374} + \frac{.758}{.302} = 4.64$$

$$S(fp) = S(fo) = 4.54$$

$$S(kol) = \frac{1.33}{.689} + \frac{1.27}{.59} + \frac{1.41}{8.42} = 5.73$$

$$S(kph) = S(koh) = 5.90$$

$$S(kon) = 6.13$$

The ordering is then,

e > mk > fo ~ fp > fm > kon > kph ~ koh > kol

which is identical to the ordering obtained from equation (2).

In summary then the exact analytical solutions of Barlow and Proschan are to be preferred if (1) either a reliability function exists, or (2) if the probabilistic expression obtained from the Boolean algebra equation for success is not too extensive. The author leaves the determination of how many intersections are too many to the individual test officer or analyst. While not aesthetically as pleasing, the sensitivity ratios of Larsen and its extension by the author appear to give corresponding results with less tedious computation for large systems.

CHAPTER IV

AN EXAMPLE

This research was originally directed toward the operational testing of automated command and control systems for use in the U.S. Army division. Consequently, it is only fitting that the example of the application of fault tree analysis to operational test design be an automated command and control system of the type currently under study. For the uninitiated an automated command and control system is essentially a management information system for use in a combat environment.

Consider a system with one central computer and one mini-computer as an operating standby for local use only. This computer center is located at division rear which has five MIODs (Message Input Output Devices) at its disposal. Additionally, there are five major headquarters forward which have one MIOD each; two division tactical command posts and three brigade command posts. These systems all have their own power sources. They are linked together by cable and VHF radio circuitry. Of course each piece of equipment, with some exceptions, has a human operator. Figure 9 is a simple schematic of such a system.

Much of what has just been done is a part of system definition. This situation will generally exist. As the

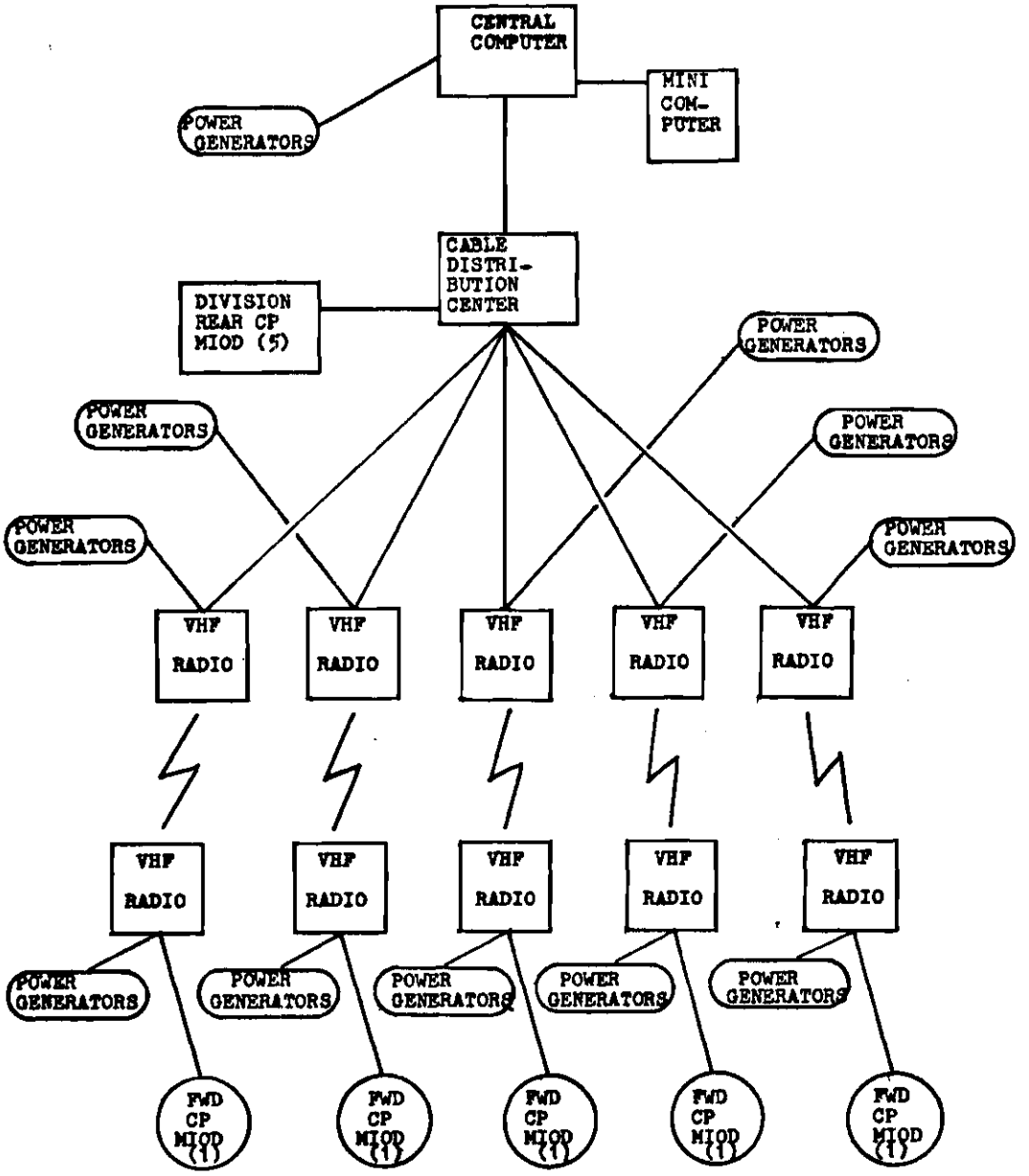


Figure 9. Command and Control Schematic

analyst becomes knowledgeable about the system, an elaborate system definition becomes unnecessary. The system boundary conditions must be included as an absolute minimun.

TOP Event: The division rear element is unable to use the computer for a length of time greater than m minutes.

Initial Conditions:

(1) The system is currently operational in all modes.

(2) No component has more than one operating state.

Not Allowed Conditions:

(1) Cable failures to include distribution center.

(2) MIOD power supply failures.

(3) Failures due to effects external to system

Existing Conditions: None

Once the analyst understands the system and has defined it, the fault tree is constructed. The fault tree is shown in Figures 10 and 11.

Notice the use of the triangles in this tree; Triangle 1 is used simply as a connecting symbol. Triangle 2, on the other hand, is a transfer symbol indicating that both CPUs are dependent on the same power source.

Compare the input and output fault events for gate 9, an OR gate, and for gate 12 an AND gate. The situations

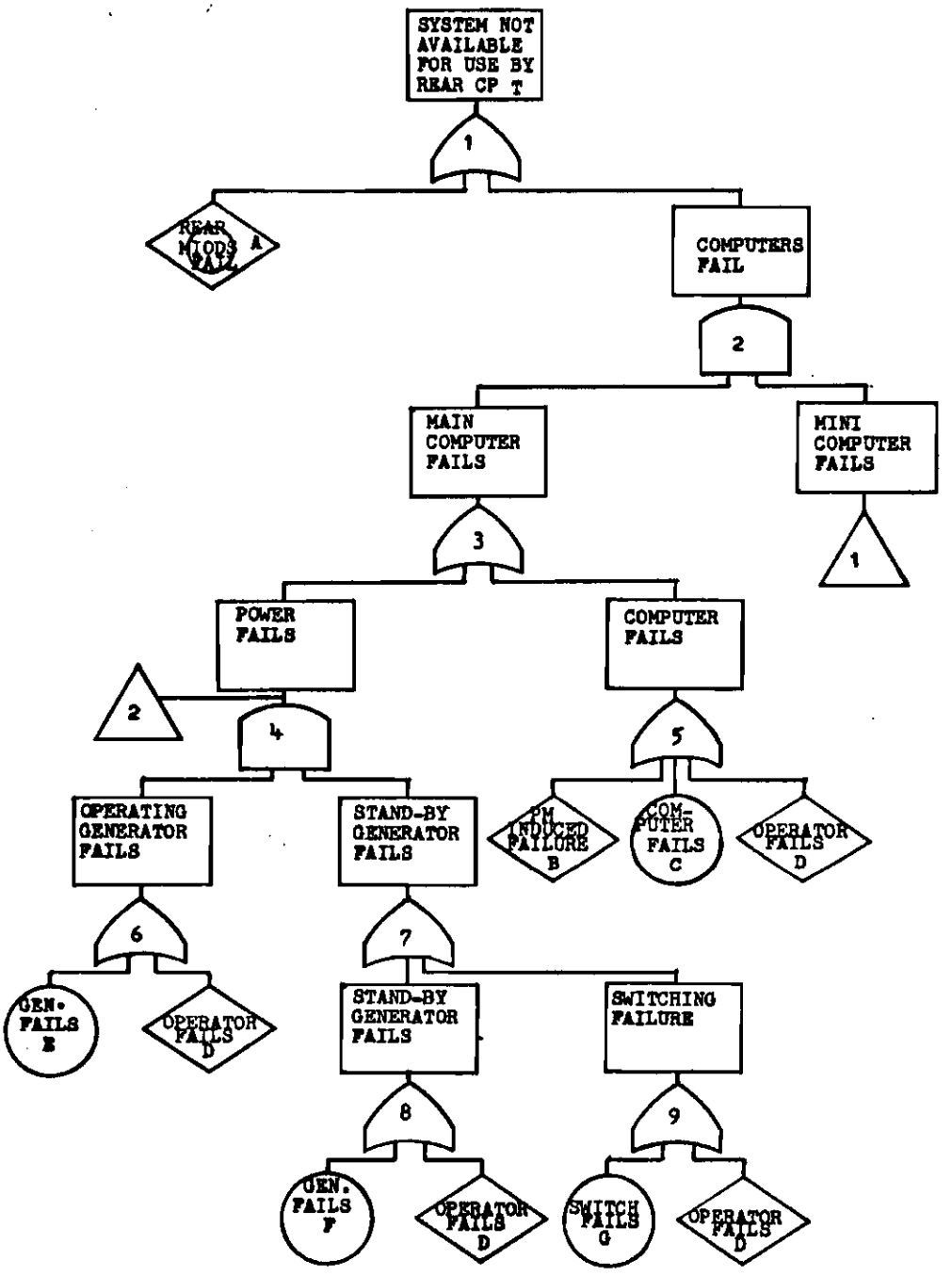


Figure 10. Fault Tree for C & C System

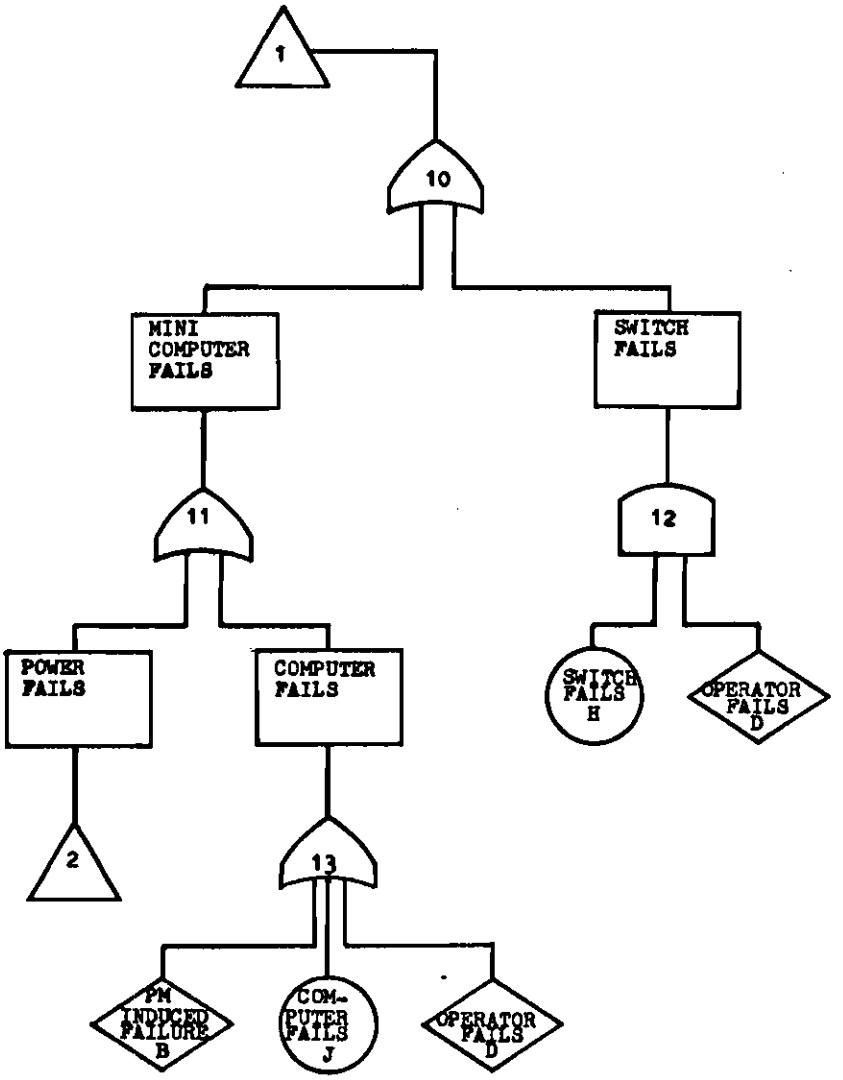


Figure 11. Continuation of Fault Tree for C & C System

are similar, yet the gates are different. The analyst considered the generator switch a manual operation. The switch giving local users access to the stand-by mini computer was considered automatic with manual override; hence, an AND situation which could have been followed with another OR gate similar to gate 9.

Recall the circle enclosed by a diamond symbol denotes an event to be treated as a component, but which is developed separately. Figure 12 shows the tree developed for component A.

Using the MICSUP algorithm the minimal cut sets for the MIOD fault tree are:

$$G_6^* = (L)(B)(G)$$

$$G_5^* = (F)(K)(B)$$

$$G_4^* = (B)(E)(J)$$

$$G_3^* = (B)(I)(D)$$

$$G_2^* = (C)(B)(H)$$

$$G_1 = G_2^* \times G_3^* \times G_4^* \times G_5^* \times G_6^*$$

then,

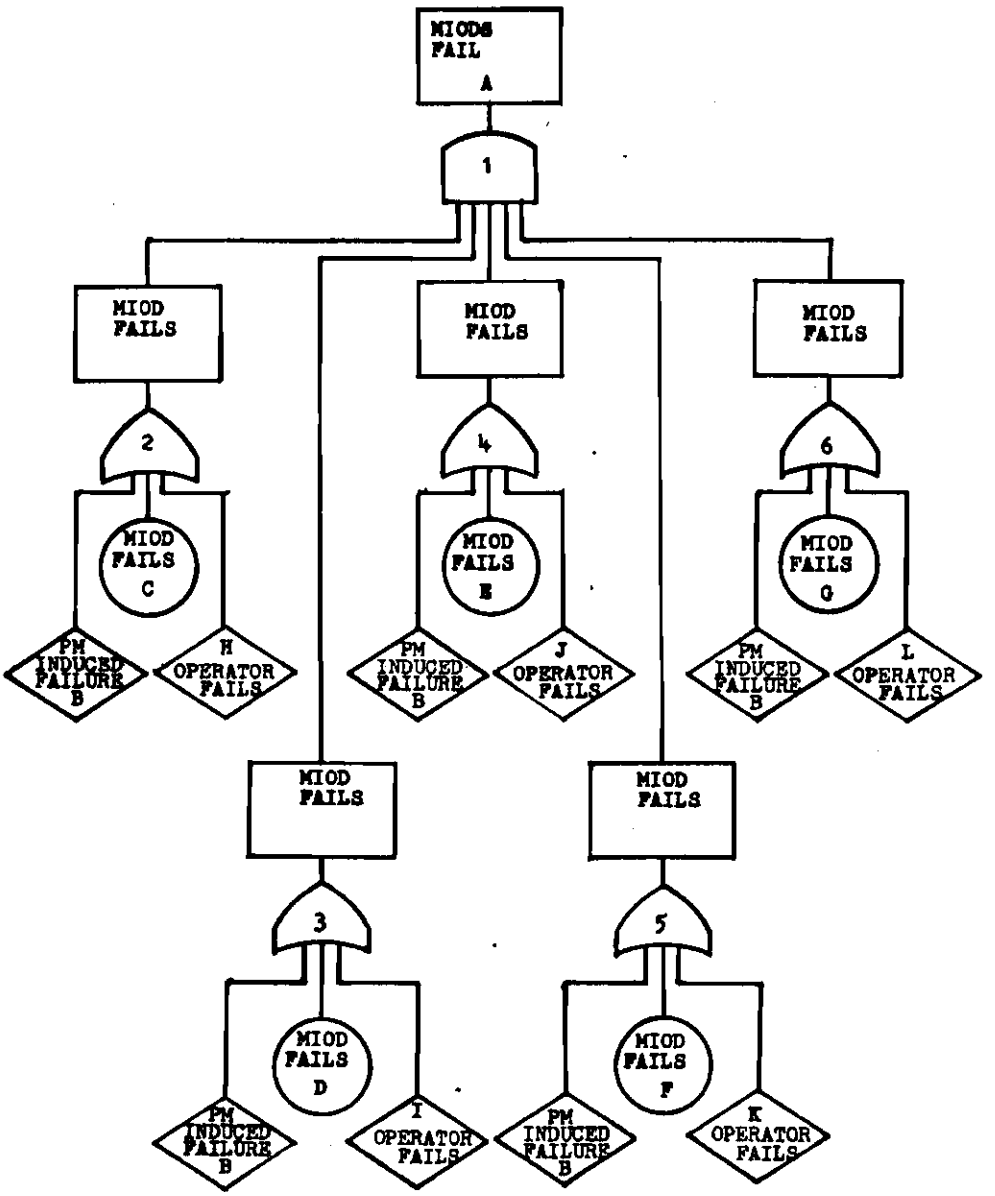


Figure 12. MIOD Failure Fault Tree

$$G_1^* = (B)(LFECD)(LFECI)(LFEHD)(LFEHI)(LFJCD)(LFJCI)$$
$$(LFJHD)(LFJHI)(LKECD)(LKECI)(LKEHD)(LKEHI)$$
$$(LKJCD)(LKJCI)(LKJHD)(LKJHI)(GFECD)(GFECI)$$
$$(GFEHD)(GFEHI)(GFJCD)(GFJCI)(GFJHD)(GFJHI)$$
$$(GKECD)(GKECI)(GKEHD)(GKEHI)(GKJCD)(GKJCI)$$
$$(GKJHD)(GKJHI)$$

From these cut sets it should be obvious why the MIOD expansion was represented by a circle enclosed in a diamond. Further analysis of this tree is really not essential to the problem. The parallel redundancy of five MIODs each with its own operator and capability of primary failure have only two possible commonalities: one could be the power source which is a not-allowed condition since sources of this type are usually plentiful in a command post. The other is a common repairman who could cause a common failure through faulty preventive maintenance, but this should be a rare occurrence. The probability that all five MIODs fail should also be as rare. Inspection of the minimal cut sets reveals their ordering as,

$$B > LFECD \sim LFECI \sim \ldots \sim GKJHI$$

and the ordering of components,

$$B > C \sim D \sim \ldots \sim L$$

Returning to the fault tree of the complete C & C system, the minimal cut sets are readily determined.

$$G_9^* = (G)(D)$$

$$G_8^* = (F)(D)$$

$$G_7^* = (G)(F)(D)$$

$$G_6^* = (E)(D)$$

$$G_5^* = (B)(C)(D)$$

$$G_4^* = (D)(EG)(EF)$$

$$G_{13}^* = (B)(J)(D)$$

$$G_{12}^* = (DH)$$

$$G_{11} = G_{13}^* \cup G_4^*$$

then

$$G_{11}^* = (B)(J)(D)(EG)(EF)$$

$$G_{10} = G_{11}^* \cup G_{12}^*$$

then

$$G_{10}^* = (B)(J)(D)(EG)(EF)$$

$$G_3^* = (D)(B)(C)(EG)(EF)$$

$$G_2 = G_3^* \times G_{10}^*$$

then

$$G_2^* = (D)(B)(CJ)(EF)(EG)$$

and

$$G_1^* = (A)(D)(B)(CJ)(EF)(EG)$$

In a similar manner by transposing the AND gates and the OR gates, the minimal path sets can be determined. Recall that the minimal path sets are the minimal cut sets determined from the dual fault tree. The minimal path sets are valuable. If all the components in any one minimal path set are operational then the system is operational. The minimal path sets for this system are:

$$\bar{G}_1^* = (ABDFGJ)(ABDEJ)(ABCDFG)(ABCDE)$$

Once the path sets are determined, the test officer can decide whether to use the exact analytical methods with h(.), the reliability function, represented by the probabilistic expansion of the Boolean algebra sum obtained from $\bar{G}_1^*$, or to use the sensitivity ratings. In this case, the analyst elects to use the sensitivity ratings. The Boolean algebra expression for the TOP event is,

$$T = A + D + B + CJ + EF + EG$$

Substituting .1 for each component yields

$$T = .33$$

Substituting .5 for each component successively and dividing through by .33 gives the gollowing ordered sensitivity ratings:

	Sensitivity Ratio	Sensitivity Rating
A,B,D	.73 ÷ .33	2.21
E	.41 ÷ .33	1.24
C,F,G,J	.37 ÷ .33	1.12
H	0	0.0

Before any attempt is made to draw conclusions from this data, determine the importance of the minimal cut sets. Using the sensitivity analysis technique the ordered results obtained are:

A,D,B		2.21
CJ	$\frac{.82}{.42} + \frac{.82}{.42}$	3.9
EF,EG		3.83

First the analyst observes that A is not in fact rated that highly. Recall A represents the failure of five MIODs in parallel. The fault tree for A was developed separately from Figure 12.

D and B are two other highly critical components which are also minimal cut sets of size one. These are both human operators. B is the computer repairman who can induce failure through faulty preventive maintenance.

PM induced error can be critical and has the tendency to stand out when the assumption is made the same repairman will make the same error on like pieces of equipment. The two actions could be handled as separate components, say B1 and B2, to decrease the emphasis. D, however, should not be passed off so lightly. This operator has several tasks, all important, which can cripple the system. The analyst might wish to include a supervisor to check D's actions or to give him an assistant to handle the power generating equipment, and then observe the impact through the tree's structure.

CJ is the most important size 2 cut set. CJ is the primary failure of the central and the back-up computer. This situation is to be expected and is more a function of computer reliability design or economics. There is little the analyst can vary and it is no great revelation that CJ is important.

Minimal cut sets EF and EG are the two least important, but the common E which has a higher rating for components should direct the analyst's attention to the power generating system. This is a definite system weakness that both computers depend on the same power generating system.

To demonstrate the effect on the system, let the stand-by computer have its own power generating sub-system identical to the other system, but separate. The new

input to (2) is shown in Figure 13.

Using the same procedures as before the new cut sets are:

$$G_1^* = (A)(D)(B)(CJ)(EGJ)(EFJ)(CE'G')(CE'F')$$
$$(EGE'G')(EGE'F')(EFE'G')(EFE'F')$$

Hence the new pessimistic estimate of the probability of failure, computed at .1, is .31, which is an improvement over .33. The new sensitivity ratings are shown below in ordered sequence.

Component	Sensitivity Rating
A,D,B	2.27
C,J	1.15
E,E'	1.03
F,F',G,G'	1.02
H	0

Minimal Cut Set	Size	Sensitivity Rating
A,D,B	1	2.27
C,J	2	3.92
EGJ,EFJ,CE'G',CE'F'	3	5.62
EGE'G',EGE'F',EFE'G' EFE'F'	4	7.30

This wiggle caused components C and J to increase in importance, as they should be. E and E' are the operating generators, hence, they are rated higher. It

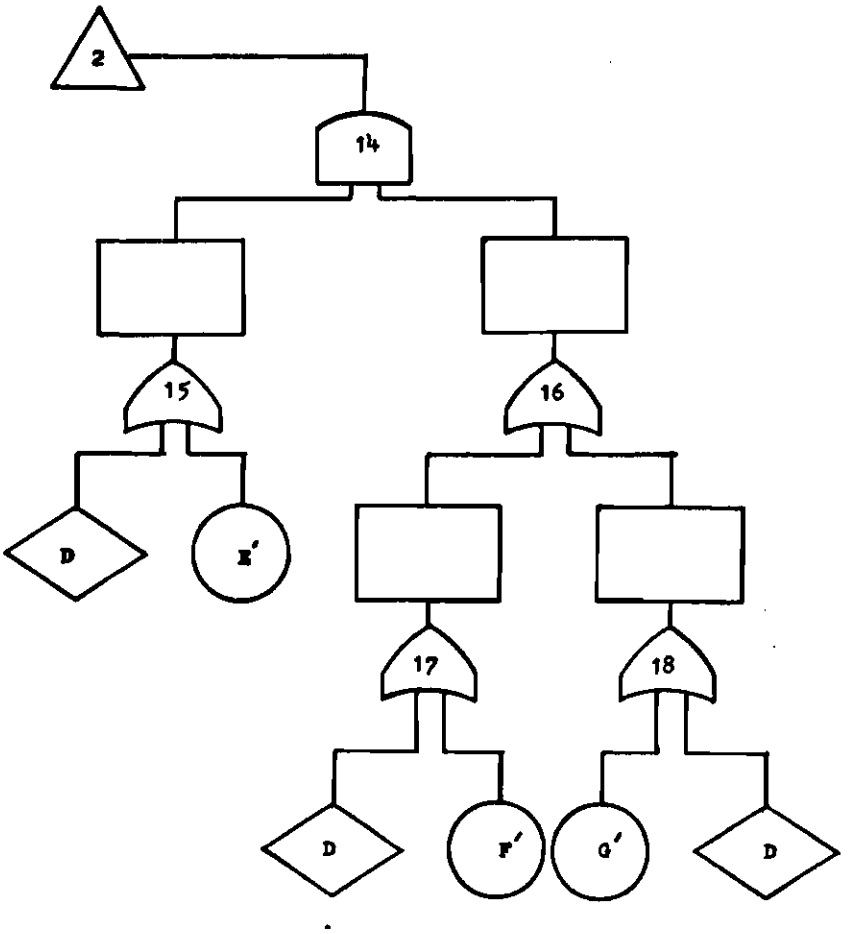


Figure 13. A Separate Power Generating System

might seem strange that all minimal cut sets of size 3 and of size 4 have the same rating. However, the size 3 cut sets represent combinations of one CPU failure and the power system failure of the other CPU. Size 4 cut sets represent various combinations of power system failures.

This example was not intended to be exhaustive. There are other TOP events that might need exploration. For instance, the probability of basing a decision on false information is one of several that comes to mind. The determination of which failures to explore is not a part of this research. The mission and the critical issues should bear heavily on the selection of the TOP events to be explored. The example does demonstrate the ability of fault tree analysis to isolate system weaknesses that might otherwise be missed in test design. It identifies the system components, that cause the failures considered; it orders them structurally without any dependence on data which may not be available; and it provides a vehicle for exploring various options that might lessen the problem. Most importantly for test design is that it aids the test planner to fuller understanding and insures that critical sub-systems and components are recognized in order that the tests will not miss their vulnerability.

CHAPTER V

CONCLUSIONS, LIMITATIONS, AND RECOMMENDATIONS

Fault trees can have a definite application to operational test design. They effectively encompass all sub-systems and components, human and hardware, of any complex system. The weakest links in the system can be revealed. This discovery is particularly important for those elements that might otherwise have escaped detection until full-scale employment. It is a method of modeling the system. Fault trees not only provide a means for exploring different configurations, but indicate which configurations need improvement. Using the methods discussed in Chapter II, the method is not data bound, but can be utilized in the very early stages of test planning.

This lack of data dependence is made possible by the use of the probability expression as the reliability function in the analytical solution. This probability expression is obtained by conversion from the Boolean algebra expression for nonoccurrence of the TOP event.

When computation is done manually and the analytical method becomes overly tedious, the sensitivity analysis technique of Larsen and this author's extension appear to yield good results. The author's extension of sensitivity

ratings to minimal cut sets is simple enough. It needs further testing. An obvious limitation is its inability to compare cut sets of different sizes. This is an area for further research.

The technique is weakened by several problems affecting all applications of fault trees. It is a form of binary modeling; thus, it does not treat partial failures. The completeness of the model is a function of the complexity of the system and the analyst's knowledge of the system. Common mode failures can weaken the independence assumption of primary events. A combat environment, a common manufacturer, common design; these and others are common mode failures.

The author firmly believes that the manual resolution of the fault tree is an excellent way for the test officer to gain greater understanding of the system. However, there is a trade-off here. For a complex system, even using the sensitivity analyses, the calculations become time consuming. The time consumption and the estimation loss may exceed the gain from manual resolution. The analytical methods exercised on the probabilistic expansion of the Boolean equation have some rather nice properties which could lead to a computer solution. For quantitative analysis, minimal cut set determination, and even construction, there are several good programs available. These are discussed in Appendix A.

In addition to those areas of interest or further research previously mentioned, two other areas appear worthy of mention. Despite the advantages of qualitative analysis in early test design, quantitative analysis does have some excellent capabilities, such as environmental impact, repair rates, etc. which should be fully explored.

Concurrent with this research, there was a project underway investigating the risk as a function of time or dollar cost of operational testing. Fault tree analysis would appear to have a definite application to this problem. The TOP event would be exceeding some time limit or dollar cost which could then be developed downward to subtests or lower, which would be represented by some general distribution.

Fault tree analysis appears to be an applicable modeling technique for obtaining better understanding of complex military operating systems. No excessive claims are made for the effectiveness of this technique as presented in this research. This research is intended rather to introduce to the test designer what may become an excellent tool and to the fault tree analyst, a new application.

APPENDIX A

QUANTITATIVE ANALYSIS

This appendix is not included in the text because it does not contribute to the main thrust of the research. That it is included at all can be attributed to the fact that much of the author's early work was in this area and primarily to the fact that it may be of interest to the sponsors of this research.

The methods of Barlow and Proschan for computing structural importance are derived from more general formula for use with general reliability functions and time dependency.

Fussell's method of Boolean manipulation to make all intermediate fault events independent, leads to a formulation for determining exact analytical solutions for time dependent systems.

W. E. Vesely in 1970 made an important advance in quantitative evaluation of fault trees using kinetic tree theory. His method computes exact time dependent failure probability information from general repair and failure distributions. The repair and failure intensities can vary with time. Computer programs using his method yield the probability of occurrence of the TOP event as a function of time, the expected number of occurrences of

the TOP event per unit time, the hazard rate, the expected number of occurrences of the TOP event during the time interval 0 to t. Similar information can be obtained for intermediate events, primary events, and minimal cut sets.[40,41]

There are several computer algorithms available to determine the minimal cut sets. MOCUS and MICSUP are two of these. The principles of each were discussed in Chapter II. The PREP codes also determine the minimal cut sets or failure modes. There are three PREP codes: deterministic, Monte Carlo, and Boolean substitution.[43]

Quantitative evaluation of fault trees is dependent upon data for the primary events. Human reliability data is scarce and all systems undergoing operational testing have a human element. However, there are 22 methods, eight major ones, for predicting human performance in a man-machine system. For a brief review of these methods, the reader is referred to Meister (1973). A complication to an application of these methods is the task analysis necessary to all of them. Detailed information may or may not be timely.

The example in Chapter IV was selected with quantitative analysis in mind. For operational testing, however, a problem does exist. The author found great discrepancies between laboratory data on new equipment and field data. A system such as the example is bound to

contain elements of both new and old equipment. There are many systems which fall under operational testing and many TOP events to be explored for which reliability data may not make sense or be needed. Restricting the application of fault trees in operational testing to a quantitative analysis based on reliability data, is in fact, selling short a potentially valuable methodology.

BIBLIOGRAPHY

1. Barlow, R. E. and Chatterjee, P., "Introduction to Fault Tree Analysis," AD-774072, 1973.*

2. Barlow, R. E. and Proschan, F., "Importance of System Components and Fault Tree Events," AD-777103, 1974.

3. Chatterjee, P., "Fault Tree Analysis: Min Cut Set Algorithms," AD-774100, 1974.

4. Crossetti, P. A., "Computer Program for Fault Tree Analysis," DUN-5508, 1969.

5. Crossetti, P. A., "Fault Tree Analysis with Probability Evaluation," IEEE Transactions on Nuclear Science, Feb. 1971, p. 465.

6. Crossetti, P. A. and Bruce, R. A., "Commercial Application of Fault Tree Analysis," Ninth Reliability and Maintainability Conference, Annals of Reliability and Maintainability, 1970, 9, p. 230.

7. Eagle, K. H., "Fault Tree and Reliability Analysis Comparison," Symposium on Reliability Proceedings-1969, Chicago, Jan. 1969, p. 12.

8. Eyestone, D., Personal Communication, Mar. 1975.

9. Field, P. F., DeGraft, W. E., and Smith, R. D., "Computer Programs for Automatically Analyzing and Drafting Fault Trees: PREP, KITT, and FTDP," AD-775661, 1974.

10. Fussell, J. B., "Fault Tree Analysis: Concepts and Techniques," Aerojet Nuclear Company, Idaho Falls, July, 1973.

11. Fussell, J. B., "Special Techniques for Fault Tree Analysis," Aerojet Nuclear Company, Idaho Falls, Mar. 1974.

---

*Indices such as AD-774072, DON-5508, etc. refer to U.S. Government reports placed on microfiche by the National Technical Information Service, Springfield, Virginia.

12. Fussell, J. B., "Synthetic Tree Model: A Formal Methodology for Fault Tree Construction," Doctoral Thesis, Georgia Institute of Technology, 1973.

13. Fussell, J. B., "Synthetic Tree Model: A Formal Methodology for Fault Tree Construction," ANCR-1098, 1973.

14. Fussell, J. B., Henry, E. B., and Marshall, N. H., "MOCUS: A Computer Program to Obtain Minimal Sets from Fault Trees," ANCR-1156, 1974.

15. Fussell, J. B., Powers, G. J., and Bennetts, R. G., "Fault Trees: A State of the Art Discussion," IEEE Transactions on Reliability, Vol. R-23, No. 1, April, 1974, p. 51.

16. Fussell, J. B. and Vesely, W. E., "A New Methodology for Obtaining Cut Sets for Fault Trees," American Nuclear Society Transactions, Vol. 15, No. 1, 1972, p. 262.

17. Green, A. E. and Bourne, A. J., Reliability Technology, John Wiley and Sons, Ltd, London, 1972.

18. Hamilton, W. F. and Nance, D. K., "Systems Analysis of Urban Transportation," Scientific American, 1969, 221, p. 19.

19. Hassl, D. F., "Advanced Concepts in Fault Tree Analysis," System Safety Symposium, June, 1965, Seattle, The Boeing Company.

20. Knauer, W., Personal Communication, Mar. 1975.

21. Lambert, H., "Fault Tree Analysis: An Overview," UCRL-75904, 1974.

22. Lambert, H. E., "Systems Safety Analysis and Fault Tree Analysis," UCID-16238, 1973.

23. Larsen, W. F., "Fault Tree Analysis," AD-774843, 1974.

24. McQuay, W. K., "Computer Simulation Methods for Military Operations Research," AD-771614, 1973.

25. Mearns, A. B., "Fault Tree Analysis: The Study of Unlikely Events in Complex Systems," System Safety Symposium, June, 1965.

26. Meister, D., "Comparative Analysis of Human Reliability Models," AD-734432, 1971.

27. Meister, D., "A Critical Review of Human Performance Reliability Predictive Methods," IEEE Transactions on Reliability, Vol. R-22, No. 3, Aug. 1973, p. 16.

28. Michels, J. M., "Computer Evaluation of the Safety Fault Tree Model," System Safety Symposium, June 1965.

29. Nagel, P. M., "Importance Sampling in Systems Simulation," Annals of Reliability and Maintainability, 1966, 5, p. 330.

30. "Operational Test Methodology Guide, Volume II, Techniques and Guidelines," U.S. Army Test and Evaluation Agency, Jan. 1974.

31. Rainy, R., Personal Communication, Mar. 1975.

32. "Reactor Safety Study: An Assessment of Accident Risk in U.S. Commercial Nuclear Power Plants, Appendix II (Volume I) Fault Tree Methodology," WASH-1400, 1974.

33. "Reactor Safety Study: An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants. Appendix III (Volume I) Failure Data," WASH-1400, 1974.

34. Report to the President and the Secretary of Defense on Department of Defense by the Blue Ribbon Defense Panel. Appendix F, Staff Report on Operational Test and Evaluation, AD-766059, 1973.

35. Semanderes, S. N., "ELRAFT, A Computer Program for the Efficient Logic Reduction Analysis of Fault Trees," IEEE Transactions on Nuclear Science, Feb. 1971, p. 481.

36. Schroeder, R. J., "Fault Trees for Reliability Analysis," Symposium on Reliability Proceedings-1970, Los Angeles, Jan. 1970, p. 198.

37. Study Report, Integrated Battlefield Control System, 1973.

38. Study Report, Tactical Operations System-Operable System (TOS^2), System Engineering Study, 31 Dec. 1971.

39. System Safety Symposium, Proceedings of Symposium sponsored by the University of Washington and the Boeing Company, Seattle, Washington, June 1965.

40. Vesely, W. E., "Analysis of Fault Trees by Kinetic Tree Theory," IN-1330, 1969.

41. Vesely, W. E., "Reliability and Fault Tree Applications at the NRTS," IEEE Transactions on Nuclear Science, Feb. 1971, p. 472.

42. Vesely, W. E., "A Time-Dependent Methodology for Fault Tree Evaluation," Nuclear Engineering and Design, 13(2), Aug. 1970.

43. Vesely, W. E. and Narum, R. E., "PREP and KITT: Computer Codes for the Automatic Evaluation of Fault Tree," IN-1349, 1970.